June 1986 U.S. $3 Canada $3.95

Just For Your Sanyo Personal Computer

soft sector®

The Monthly Magazine For Sanyo Personal Computer Users

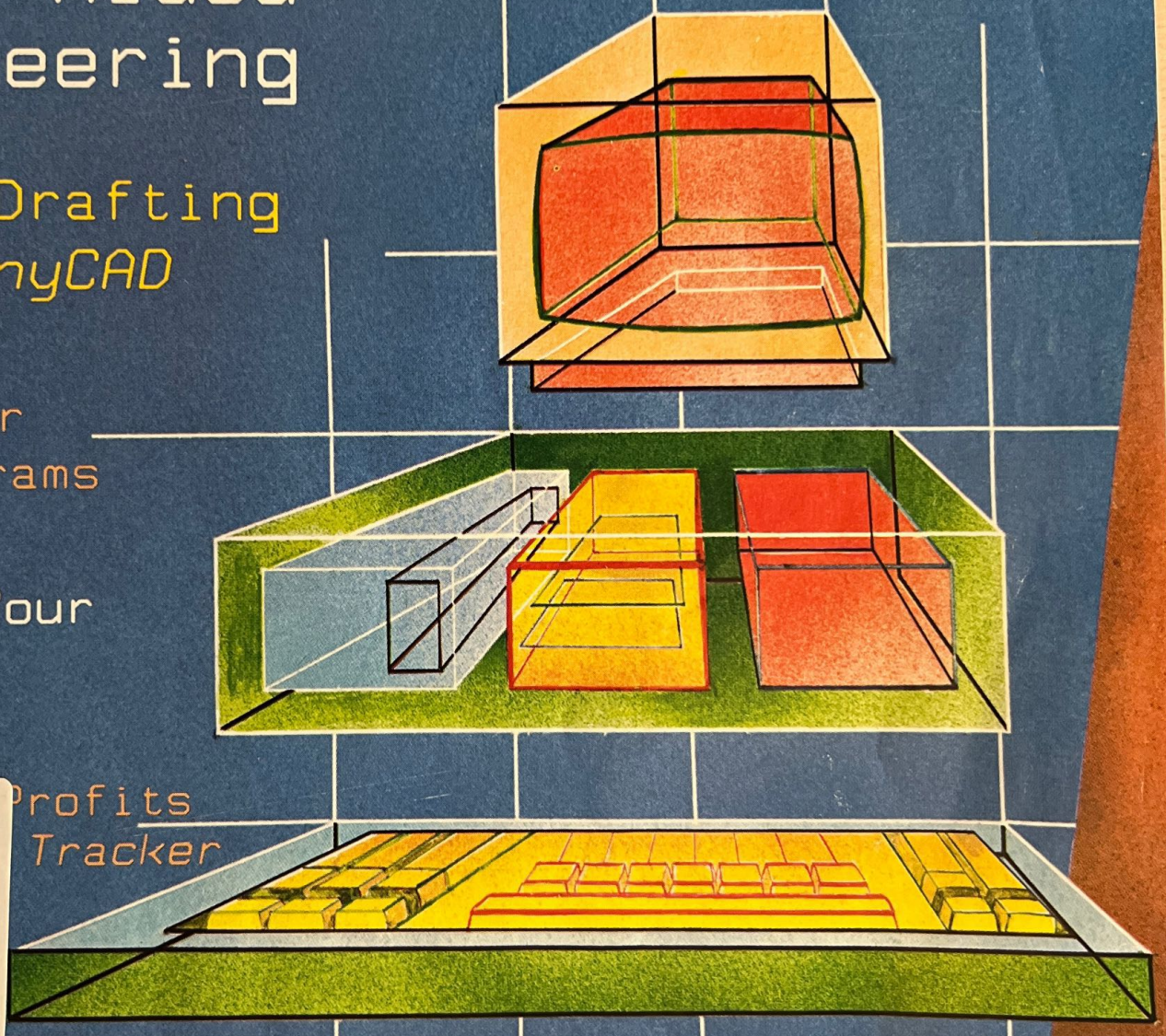
Sanyo-Aided Engineering

Easier Drafting with *TinyCAD*

A Menu for All Programs

Analyze Your *Car Cost*

Profits *Tracker*



ictory in *Othello*

us: Design Musical Creations

PICASSO

The **ULTIMATE** In Computer Graphics Programs

PICASSO by Bill Dunlevy

This is the one to set the standards! PICASSO does more than many graphics programs costing thousands of dollars more. Never before has such power and simplicity been available to the user.

What does PICASSO do? Everything!

* Large, helpful picture menus, eliminate the need to memorize control keys or cumbersome graphic languages.
* Keyboard, Joystick, or even Graphic Tablet controlled, in "directional" or "free-hand" format.
* You can choose from 64 pre-mixed colors, or mix your own for over 10,000 possible patterns, and save your favorites to a palette on disk!
* Over 20 different "brushes" let your draw with symbols in different "densities" for an air-brushed effect.
* Special "Rubber band" graphics let you see what you're doing AS you make adjustments on lines, rays, boxes, circles, polygons, stamps, etc.
* Versatile "Fill" permits you to change multi-color patterns.
* You're not restricted to simple circles and elipses here! You can create ANY kind of polygon: from triangles, to eliptical pentagons, to perfect circles and everything in between!
* Powerful "Stamp" system lets you copy, move, save, and perform complexed operations with any sections of your pictures.
* You can use your picture files with MichTron's new FREEZE FRAME graphic printing utility for fantastic hardcopy pictures, and in your own BASIC or machine language programs for all your graphic needs.

What DOESN'T PICASSO do? Restrict your imagination.

PICASSO is guaranteed to knock your smocks off!

256K SANYO MBC-550/555 required

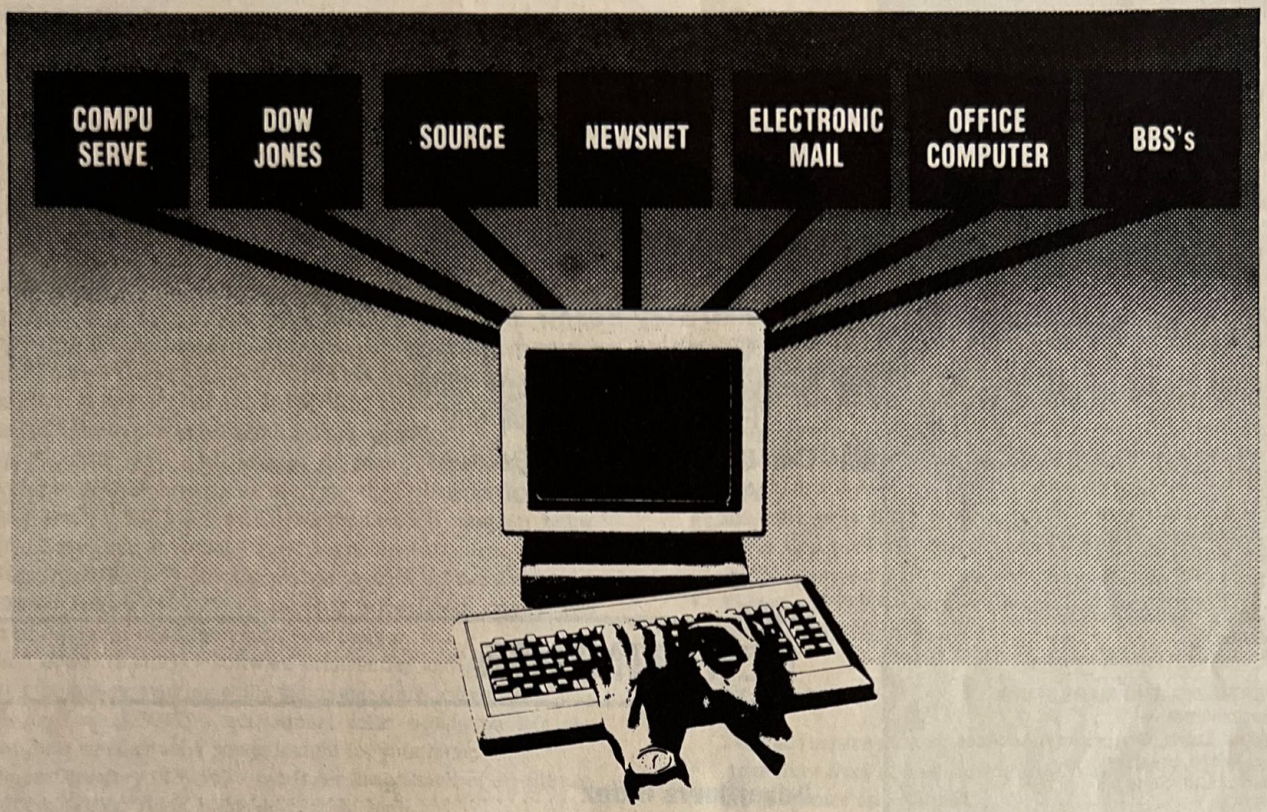
(Joystick recommended) **$99.95**

MichTron

576 S. Telegraph Road Pontiac, Michigan 48053

Orders & Info: (313) 334-5700

INTELLICOM



A COMPLETE Communications System

Now you can tap any of the countless telephone data based INFO sources, EXTRACT an FILE that data away for use later. Suddenly, your computer is the smartest terminal around because INTELLICOM emulates such terminals as:

- **DigitalVT52/VT100**
- **TeleVideo 910 • ADM3A/5**
- **ADDS Viewpoint/25**
- **CompuServe Executive**
- **Simple TTY**
- **TeleVideo 925**

Now you can transfer **both binary and text** files, using simple ASCII, ASCII Capture, Xon/Xoff, XMODEM (both Cheksum and CRC are supported) or CompuServe A Protocols. Operate at any speed up to 4800 (Sanyo) or 9600 baud (IBM PC) without ever being outdated as higher speed/lower priced modems are introduced.

Simple menu selections, with a built in **HELP** facility keep it easy forever.

And our expert mode will allow you to even bypass the menus. **Autodial directories (Hayes or equivalent modems) are supported along with simultaneous printing.** Our **built in DOS WINDOW** will actually **let you run another program** from within INTELICOM **(192K min. ram).**

Additional features: Variable Buffer Size reduces disk activity, while utilizing RAM capacity. BATch file utility program included. **Persistence feature for autodialing.**

Auto log-on script files can be used to support almost any smart modem and/or automatically log into your favorite remote system. **User defined function keys. Color support. Michtron and AOK DOS support** and more! Best of all, the entire versatile INTELLICOM package costs you only **$89.95 COMPLETE! RUSH $15 today and you'll receive a fully functional demo disk that lets you check it all out.** If you order INTELICOM, simply return the demo disk and we'll deduct your $15 from the $89.95 purchase price.

For $10.00 more we will supply you with the host support code for the error checking protocol on Dec system 10, 20, or VAX ... and save your company a bundle!

DON'T DELAY!
ORDER TODAY!
RETURN UPS SHIPMENT GUARANTEED! HURRY!

COMPUTER TOOLBOX inc.

MasterCard/VISA/COD orders accepted. Specify computer and DOS version when ordering. Connecticut residents add 7½% sales tax. **Add $5 shipping for all orders.**

1325 East Main Street, Waterbury, CT 06705
In Connecticut call: 597-0273

soft sector

The Monthly Magazine for Sanyo Personal Computer Users

Vol. II, Issue 11 June 1986

Editor and Publisher
Lawrence C. Falk

Managing Editor Ed Ellers
Senior Editor Tamara Dunn
Consulting Editors Gordon Monnier, T. Kevin Nickols
Submissions Editor Jutta Kapfhammer
Reviews Editor Monica Dorth
Editorial Consultants Jo Anna Arnott, Danny Humphress, Belinda C. Kirby
Contributing Editors Fred Blechman, Brian M. Stone, Charlotte A. Stone
Technical Assistants Cray Augsburg, Chris Wehner
Editorial Assistants Wendy Falk, Judi Hutchinson, Angela Kapfhammer, Shirley Morgan

Art Director Sandra Underwood
Design Staff Jody Gilbert, Tracey Jones, Heidi Maxedon, Kevin Quiggins
Production Assistant Cindy Jett

Chief of Typography Debbie Hartley
Typography Services
Jody Doyle, Suzanne Benish Kurowsky

ADVERTISING AND MARKETING
Western U.S. Representative
Shackleford, Nolan, Davis, Gregg and Associates
Cindy J. Shackleford, president
12110 Meridian South, Suite 5
P.O. Box 73-578
Puyallup, WA 98373-0578

Advertising Coordinator Doris Taylor
Advertising Representative Kim Vincent
Advertising Assistant Debbie Baxter
(502) 228-4492

For Soft Sector Advertising and Marketing office information, see Page 62.

FPSS, Ag. Publications Enterprises, Inc.

President Lawrence C. Falk

General Manager Patricia H. Hirsch
Asst. General Mgr. for Finance Donna Shuck
Admin. Asst. to the Publisher Sue Rodgers

Editorial Director James E. Reed
Asst. Editorial Director Jutta Kapfhammer
Creative Director Jerry McKiernan

SOFT SECTOR — The Monthly Magazine for Sanyo Personal Computer Users (ISSN 8755-7460) is published every month of the year by FPSS, Ag. Publications Enterprises, Inc., The Falsoft Building, P.O. Box 385, Prospect, KY, 40059. Phone (502) 228-4492. **SOFT SECTOR — The Monthly Magazine for Sanyo Personal Computer Users, SOFT SECTOR ON DISK** and the **SOFT SECTOR** logotypes are registered ® trademarks of FPSS, Ag.

Second Class Postage Rates are paid at Prospect, Kentucky and additional offices. USPS 741-750. POSTMASTER: Send address changes to SOFT SECTOR, P.O. Box 385, Prospect, KY 40059. Forwarding Postage Guaranteed.

Subscriptions to **SOFT SECTOR — The Monthly Magazine for Sanyo Personal Computer Users** are $28 per year in the United States. Canadian rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only. Full refund after mailing of one issue. A refund of 10/12ths the subscription amount after two issues are mailed. NO refund after mailing of three or more magazines.

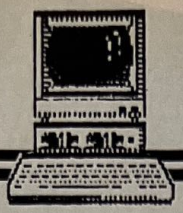
CONTENTS

FEATURES

DEPARTMENTS

REVIEWS

The small disk symbols appearing beside features and regular columns indicate that the program listings with those articles are on this month's SOFT SECTOR ON DISK, ready to LOAD and RUN. For full details, see the SOFT SECTOR ON DISK ad on Page 27.



Bringing you our very best...

With 48 programs (and still counting) we have the biggest, brightest selection of software available from any company. All our programs run on the Sanyo 550 series; those listed with an asterisk (*) also run on the Sanyo 700 and 800 series and other IBM-compatible computers.

For everyone:

BBS ($99.95) - bulletin-board system for your Sanyo.
Cornerman ($39.95) - sidekick for your Sanyo has calendar, notepad, calculator, dialer, even an ASCII table.
DI ($24.95) - advanced directory system sorts all files, even in subdirectories or on hard disks. Locates any file instantly.
* ***M-Disk*** ($34.95) - classic RAM disk emulator makes your Sanyo super-fast at any disk-addressing operation.
Mi-Key ($34.95) - a key-change program. Type long strings with two or three keystrokes, or try out the Dvorak keyboard.
* ***Mi-Term*** ($79.95) - terminal communication program helps your Sanyo talk to virtually any other computer system.
* ***Mousetick*** ($39.95) - trick to make a joystick act like a mouse.
Printer Helper ($29.95) - controller helps Epson printers use their full range of features with your Sanyo.
* ***Soft Spool*** ($34.95) - software print buffer and spooler keeps your computer working while the printer chugs along.
Solar Sim ($29.95) - educational solar simulation lets you estimate whether solar power is right for your home.
* ***SuperDirectory*** ($39.95) - cataloger keeps track of all files on all your disks. Sorts, prints disk labels, more.
Type Right ($29.95) - training program makes you a faster, more accurate typist, helps prevent frustrating typos.

For businessmen:

* ***Business Agreements*** ($49.95) - attorney-prepared legal forms and contracts simplify your business needs.
* ***Calendar*** ($24.95) - appointment book brings order to your busy life at work or at home.
MasterGraph III ($89.95) - business graphics package creates dramatic line, bar and pie graphs.
* ***Mi-Mail*** ($79.95) - menu-driven business mailing system creates mailing labels, even a personal phone book.
* ***Personal Money Manager*** ($49.95) - home financial program keeps track of your budget, projects fixed expenses.
* ***Pledge*** ($199.95) - tracks pledges and special gifts to religious institutions, reports to givers and the treasurer.
* ***Quick and Simple*** ($49.95) - menu-driven, down-to-earth list manager features easy input, several output options.

For graphics artists:

Grafiti ($39.95) - an easy-to-use, basic drawing program.
Picasso ($99.95) - menu-driven, advanced free-form drawing package gives you absolute control of graphics.
FreezeFrame ($39.95) - versatile screen dump utility can be customized to take full advantage of your printer.

For programmers:

* ***Drive Timer Plus*** ($24.95) - utility checks disk drives, encodes files and provides menu-driven DOS file commands.
DS DOS Plus 2.11 ($49.95) - MS-DOS enhancement package supports 80-track drives, IBM graphics, sorted directories.
* ***EasyRecord*** ($199.95) - C-programmer's file utility manipulates files of all data types with easy-to-use functions.
* ***EasyWindow*** ($199.95) - screen display manager makes custom windows easy from C-language programs.
Graphpac ($49.95) - package of graphics routines for Pascal, C and machine language programs.
Super Zap ($49.95) - full-featured disk editor lets you change anything anywhere on a disk. Fix disk errors with ease.
* ***Transfer*** ($59.95) - data transfer program converts TRS-80 files to MS-DOS files, or vice-versa.

For gamers:

Text adventure games:
* ***Adventure Disk #2*** and ***Adventure Disk #3*** ($34.95 each) - five unique games on each disk.

Arcade games:
Cashman ($34.95) - take the money and run!
Darts ($34.95) - toss darts with the joystick in eight versions of the classic game of skill.
Demon Seed ($34.95) - first bats, then demons to defeat.
Major Motion ($29.95) - save the weapons van from attack.
Maz ($24.95) - find you way through the maze without being eaten by the prowling cat.
Mudpies ($34.95) - throw them at angry circus clowns, duck the clowns and their Indian clubs.
Robounce ($24.95) - robot aliens move through invisible force fields that deflect your shots.
Speed Racer ($29.95) - win the race - or at least survive!
Thunder Chief ($34.95) - destroy enemy ground forces as you fly the Thunder Chief on increasingly difficult missions.
Time Bandit ($39.95) - collect the treasures of the ages from 20 worlds, fifteen levels of play.

Strategy games:
Checkmate ($39.95) - match wits with the chess computer.
* ***DC-10*** ($39.95) - realistic instrument flight simulator, complete with emergencies.
Emperor ($34.95) - save the Roman Empire from the barbarians, and beware of your own generals.
* ***FlipSide*** ($34.95) - try Reversi against live or computer foes.
King Arthur ($24.95) - your foresight and strategy could save Britain from invading Anglo-Saxons and Jutes..
Solitaire ($34.95) - and cribbage and blackjack and poker squares and klondike. Many hours of entertainment.
Tic Tac Toe and ***Cryptogram*** ($29.95) - two classics.

Dealer inquiries welcome • Visa and Mastercard accepted • Add $3.00 shipping and handling to each order

MichTron

576 S. Telegraph, Pontiac, MI 48053
Orders and Information (313) 334-5700

Protect Your Valuable Magazine Collection With . . .

DISTINCTIVE AND DURABLE SOFT SECTOR BINDERS

Do yourself a favor! Protect and showcase a valuable resource — each and every issue of SOFT SECTOR — with high-quality, blue vinyl binders with the magazine's name embossed in gold.

Spend more time with your Sanyo and less on frustrating searches for misplaced issues. Organize your SOFT SECTOR library!

These handsome royal blue binders cost just $7.50 (please add $2.50 for shipping and handling per binder).

SAVE! Buy four or more back issues of SOFT SECTOR with this order and save $1 per issue. Use the Back Issue Order Form on Page 59.

Mail to:
Soft Sector Binders
The Falsoft Building
P.O. Box 385
Prospect, KY 40059

To order by phone (*credit card orders only*), call (800) 847-0309, 8 a.m. - 5 p.m. EST. All other inquiries call (502) 228-4492.

☐ **YES. Please send me** _____ **SOFT SECTOR binder(s)** at $7.50 each (plus $2.50 for shipping and handling per binder). If your order is to be sent via U.S. Mail to a post office box or to another country, please add $2. Kentucky residents add 5% sales tax.

Name _____

Address _____

City _____ State _____ ZIP _____

☐ My check in the amount of _____ is enclosed. (In order to hold down costs, we do not bill.)

Charge to: ☐ VISA ☐ MasterCard ☐ American Express

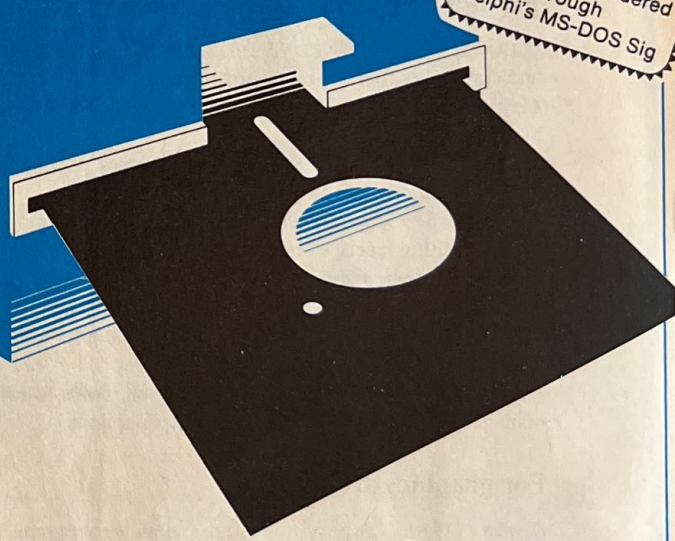
Account Number _____ Exp. Date _____

Signature _____

For your convenience, these items may be ordered through Delphi's MS-DOS Sig

SOFT SECTOR ON DISK

The cheaper, faster companion method to Soft Sector programs!



Each and every month of the year, SOFT SECTOR offers an assortment of programs for the home and office — utilities, business applications, games and more — all for the price of typing in the listings. But for the person on the move, that's a high price to pay.

BACK ISSUES AVAILABLE

Several of the first issues of SOFT SECTOR are now out of stock, but the programs from those issues are available as back orders of SOFT SECTOR ON DISK, along with the accompanying text files of the articles ready to be routed to your screen or printer with a simple TYPE command. This may be your only chance to get the programs that you've missed.

Volume I contains programs from August, September and October 1984; Volume II contains programs from November and December 1984 and January 1985. All other copies of SOFT SECTOR ON DISK include programs from individual issues of SOFT SECTOR, and should be used in conjunction with the accompanying issues of the magazine.

Now is your chance to make your Sanyo a full-time computer instead of a typewriter. Subscribe to SOFT SECTOR ON DISK today and give those weary fingers a break! Look for the subscription card between pages 16 and 17.

SOFT SECTOR ON DISK subscription rate is: within the U.S., $99; Canadian rate, U.S. $115; all other countries, U.S. $130. SOFT SECTOR ON DISK single volume rate is: within the U.S. $12; Canadian rate, U.S. $14; all other countries, U.S. $16. Subscription orders begin with the current issue, allowing 6 to 8 weeks for first delivery. **U.S. currency *only*, please.**

To order by phone (*credit card orders only*), call (800) 847-0309, 8 a.m. - 5 p.m. EST. All other inquiries call (502) 228-4492.

SOFT SOAPBOX

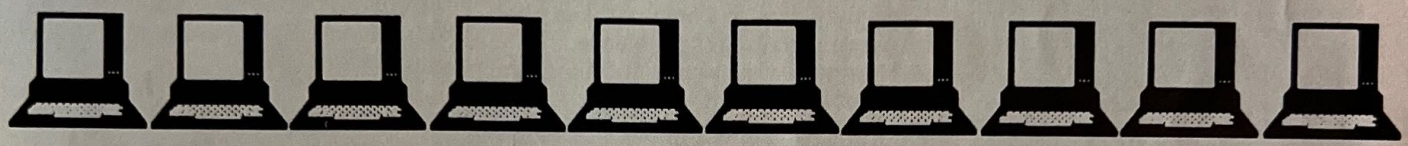
This month's Soft Soapbox deals with something that's here, something that's coming and somebody who's leaving.

The something that's here is *TinyCAD*, which you'll find on Page 20. It is a very nice computer-aided drafting package for the Sanyo 550 that was written by an experienced draftsman for his own use. Ron Smith is a consulting mechanical designer at the Lawrence Livermore National Laboratory, and he has obviously put lots of effort into this program. *TinyCAD* has some features that I've only seen in some very expensive CAD packages for PC compatibles, and a zoom capability that must be seen to be believed. Engineers these days are going wild over CAD systems costing $10,000 and up — way up. *TinyCAD* could very well be an answer to the drafting needs of a number of people whose work involves fairly simple drawings.

The something that's coming is the MBC-990, Sanyo's long-awaited answer to the IBM Personal Computer AT. This machine runs at 8 MHz with no wait states (unlike the original PC AT, which ran at 6 MHz and used one wait state) and, therefore runs as fast as any "showroom stock" MS-DOS personal computer around. (Perhaps it's the Dodge Omni GLH of the PC compatible world?) If all goes well we should have more to tell you about the 990 in the July issue.

On a final note, somebody is leaving SOFT SECTOR. Monica Dorth, who has been involved with this publication almost since its inception, is leaving us. As reviews editor, Monica has built up a "fleet" of reviewers that has helped us — and our readers — immensely. Monica has been a real credit to the Patrol; she will be missed.

Ed Ellers



LETTERS TO THE EDITOR

PAR AVION

Memory Upgrade Hints

Editor:

I am writing to thank Bob Jack for yet another excellent article, "Piggybacking" (April 1986, Page 20). I would like to suggest some do's and don'ts that your readers may need to know to avoid problems when installing the upgrade.

- Do test all the chips in the expansion RAM sockets before soldering a stack together. I had two bad chips, one on each layer; this disabled the expansion beyond 256K for two combinations of CAS.
- Don't use wirewrap wire for the Pin 15 interconnections! It is great for on-board jumpers where there is little stress, but the hanging leads and the unexpected debugging caused several breaks which gave me fits.
- Don't use stranded wire, because those pesky stray strands lead to erroneous and inconsistent interconnections. I would recommend #26 insulated solid wire.
- A neat way to pull up the various lines is with a resistor pack which can be glued right on top of, or near, the 74LS138. All leads are then short and neat. The +5V source is available on Pin 6; the other signals are at 7, 9, 10 and 11. The resistor pack leads are good tie points for the signals to Pin 15 on the expansion RAM chips.
- Remember to install the power supply modification *before* doing any memory expansion at all. It should be tested with the original configuration so that the question of an inadequate +5V supply doesn't confuse the issue of whether the memory expansion works.
- Very carefully re-mate all connectors before applying power after the modifications. It is possible to do some serious damage if this precaution is not observed.
- Do seriously consider adding a defeat switch to disable the expanded memory! Some programs will not work with the added RAM or with modified versions of MS-DOS. I chose to connect an SPST switch between Pin 11 of the 74LS128 and the 1K pullup resistor; the resistor must remain connected to Pin 15 of the memory chips at all times.

Keep up the good work!

Joe Sartori
Mission Viejo, CA

SSOD with GW-BASIC

Editor:

Many of the files on SOFT SECTOR ON DISK ending in .BAS will not run using the Video RAM Board and GW-BASIC. What am I doing wrong?

Henry Herman
Woodstock, GA

Editor's Note: Most of the programs in SOFT SECTOR and on the disk are intended for use with the standard Sanyo 550 series machines, and are not intended to be used with the Video RAM Board.

Invading the Public Domain

Editor:

I'm fairly new to computing and am obtaining a lot of public domain software over the telephone lines, which brings me to the purpose of this letter. I would like to suggest a public domain department as a monthly addition to SOFT SECTOR; it could review software, sources, bulletin boards and other topics. A lot of the software I download is squeezed or archived with programs I don't understand. A large portion of it comes without documentation, leaving users like me unable to understand its use. I feel that a column on the public-domain world would be most helpful to me and many of your readers.

George Patterson
Salem, OR

Editor's Note: Our monthly Delphi Bureau column frequently discusses public domain software that can be downloaded from Delphi's MS-DOS SIG.

Keyboard Extension Revisited

Editor:

I find that the keyboard cord for the Sanyo is like a stiff neck (no maneuverability), so it was to some delight that I read John Kelty's "Make Your Own Keyboard Extension Cable" (May '85, Page 87). I made a couple of changes, though, in that I used a 6-foot cable with 5-pin DIN plugs at each end (Radio Shack catalog number 42-2151) and a 5-pin inline DIN socket (274-006). These two items total $6.48 (U.S. prices). One of the plugs was removed and replaced by the inline socket, and after managing to mix up the wires I got it working in about half an hour.

Ed Howell
Lower Sackville, Nova Scotia

MS-DOS 2.11 with 160K Drives

Editor:

As an owner of an MBC-555 with single-sided disk drives, I turn green with envy and burn with frustration when I see that many of the programs in your grand magazine will run only with MS-DOS 2.11. Is there any way I can overcome this problem? The Sanyo agent here tells me that I can't run MS-DOS 2.11 on my single-sided system.

Neale R. Towers
Hamilton, New Zealand

Editor's Note: If you don't want to install double-sided disk drives, you can format an MS-DOS 2.11 system disk that will run on a single-sided machine. Boot DOS 2.11 on a Sanyo 555 with double-sided drives, place a blank disk in Drive B and type the command `FORMAT B: /1 /S`. *This will format the diskette on one side only and copy the system to it. The resulting DOS 2.11 system disk should work fine on a single-sided Sanyo 550, but remember to add the* `/1` *to the* `FORMAT` *command when formatting diskettes under 2.11.*

PC BASIC Conversion Hints

Editor:

Regarding the article "Sanyo BASIC to PC BASIC Converter" (April 1986, Page 10), I would like to add my two cents worth to improve an already useful program. I suggest the following changes:

- Delete line 100.
- Add this line: `145 BYTES = LOF(1)/128`
- Change `BYTES/128` to `BYTES` in lines 150 and 200.

This will automatically calculate the file size of each program you convert.

Michael A. Nohrden
Des Moines, WA

Coincidences

Editor:

If possible, could you please publish a program which is able to make the keyboard a synthesizer, and some sort of memory which remembers the notes pressed?

J. Bowtell-Harris
Enoggera, Australia

Editor's Note: It just so happens that we have such a program in this issue! Check out "Sanyo Synthesizer" on Page 50.

InfoStar Patch

Editor:

A small law journal that I am connected with requires regular and extensive mailings to prospective authors. Some of the correspondence consists of first-contact form letters, some of personal inquiries or responses. Until this year, letters, names and addresses collected by previous staffs had collected in various files and boxes and paper bags, and the utility of all this information had declined gradually to near nothing.

This year, we began to enter this stuff into a file using the *InfoStar* software and MS-DOS 2.11 that came with my MBC-555, and I wrote a set of reports and `*.BAT` routines that protect against duplicate entries, automatically update known addresses from a permanent file, format address data for *MailMerge*, etc.

Everything was working fine until this week, when the system suddenly began refusing to read data file output that had been indexed using *FormSort*; *ReportStar* would give an "Error #16" message, claiming that either the key length was wrong, or the `*.NDX` file not properly terminated. This was a serious problem, since the system goes through this process a dozen times in some of the `*.BAT` routines.

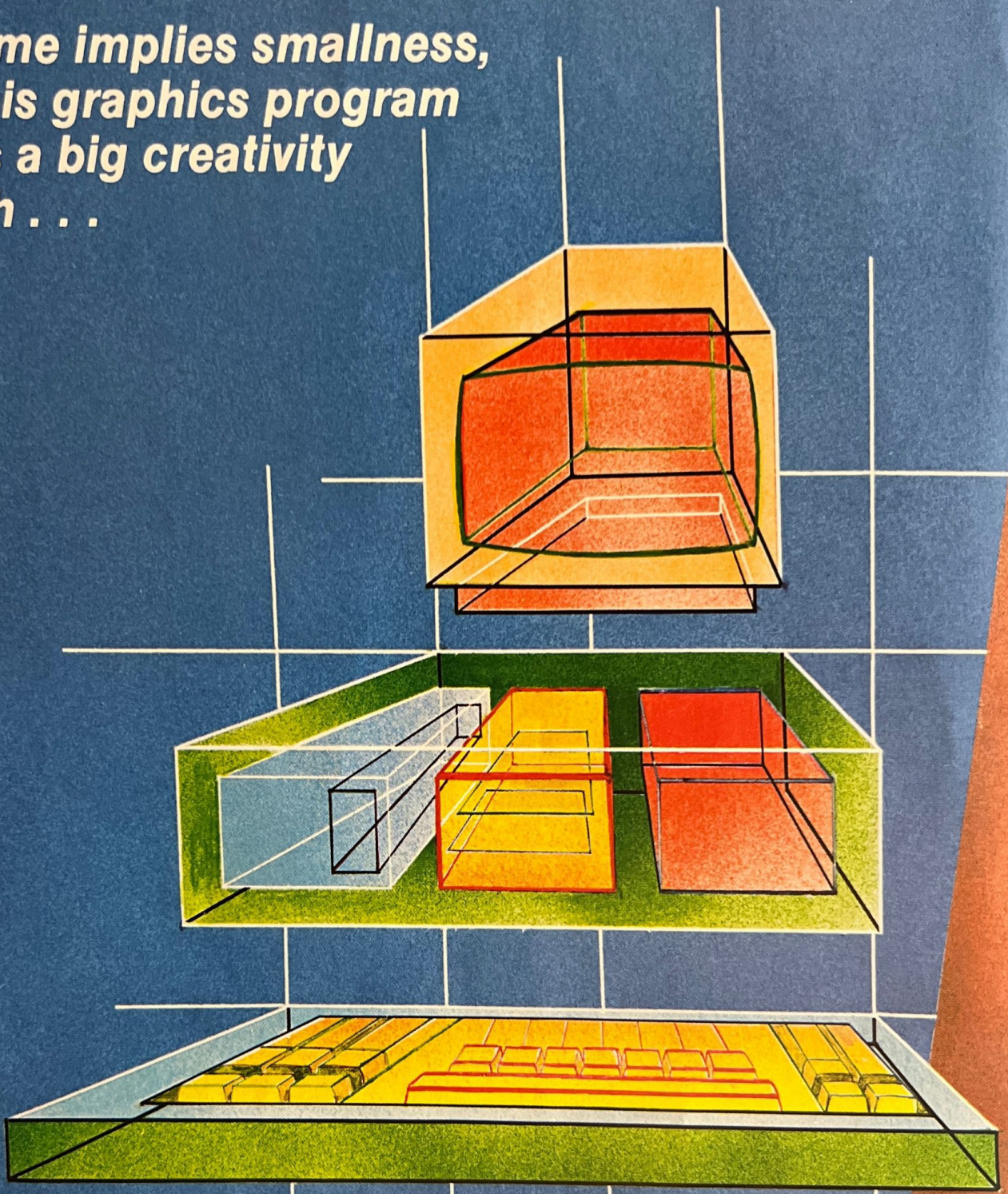
Several hours of tampering revealed only very awkward ways to getting around the problem. In exasperation, I called MicroPro in San Rafael. The *InfoStar* support department answered my question without a moment's hesitation. The patch to solve the problem turned out to be simple. If it is not common knowledge, it should be:

1) Collect *DEBUG.COM* and *FORMSORT.OVR*
2) Enter `DEBUG FORMSORT.OVR`
3) At the prompt, enter `A099E`
4) Enter `10`
5) Enter `W` to write the revision
6) Enter `Q` to return to system

I just recently discovered your publication. I hope you keep those back issues in print; seems I have a little reading to catch up on.

Frank Bennett, Jr.
Los Angeles, CA

Its name implies smallness, but this graphics program packs a big creativity punch . . .



TinyCAD

By Ron Smith

TinyCAD is a two-dimensional graphics program designed for the Sanyo MBC-550 computer. It can be used as a general design aid or can be customized to suit particular needs.

The program uses the exceptional graphics capabilities of the Sanyo computer and Sanyo BASIC. There is no need for a mouse, light pen, digitizer pad or joystick (although they may be integrated if you wish). Instead, we use the graphics cursor (GCURSOR) in BASIC and the cursor and tab keys on the console.

Written for simplicity and flexibility, *TinyCAD* leaves an open door to creativity. It allows for the separate design of an algorithm and simple merging into the program body. It lets you design your own designing tool. For example, let's assume you have interests in the architectural area of CAD. You may wish to design a set of algorithms, such as electrical outlets, lighting fixtures, sinks and bathtubs, and merge them into the program. *TinyCAD* readily accepts this.

Included in this program is a minimum set of algorithms — a starter set so to speak. However, I can assure you there are many prospects for expansion. As an example, there are now more than 30 algorithms created for this program, all of which were designed and debugged externally and all of good quality.

I resisted the temptation to clutter this program with error-trapping routines. *TinyCAD* is designed to be easily coded from text, and extra bells and whistles simply add lines of confusing code. In any case, the user is free to add as he/she sees fit. The same logic applies to the somewhat cumbersome file-handling methods. However, ease of algorithm transfer is the primary consideration.

Walk through the System

You may wish to keep the coding to a minimum and enter only the program body and one or two algorithms. If such is the case, look through the program listing; it should be obvious which lines to skip. When finished with the coding, you must take care of one file before running. The FILELIST has not been initialized. If you are not familiar with opening disk files try this: RUN 1100. At the measurement prompt press BREAK. Now run the program. Add a new filename (up to eight characters — no extension).

Unit of Measurement — For measurement purposes, the program needs to know how many of the world coordinate positions you want to use to represent an inch, foot, meter or whatever. Suggestion: enter 100. You will get a better feel for this later. There is a slight delay while the files are set up.

Drawing Screen — Cursor top left <CR>, cursor right <CR>, <CR> selects the drawing screen from the world system and draws a screen representative rectangle around it.

Functions

OL — Object Line: Draws line from cursor position to cursor position. Line lengths are constant on the world system.

1) CURSOR TO BEGINNING <CR>, CURSOR TO END <CR>
2) OPTION: <CR> exits
3) CURSOR TO END <CR>, back to '2'

LL — Leader Line: Same as OL, different color.

CL — Construction Line: Same as OL, different color, current screen only, does not record.

OC — Object Circle: Drawn clockwise from "begin degrees" to "end degrees" with 0 and 360 at far right of circle/ellipse. Circle/ellipse sizes are constant on the world system.

1) CURSOR TO CIRCLE CENTER <CR>
2) OPTION: <CR> exits
3) CURSOR TO RADIUS <CR>
4) OPTION: Enter F for full circle
5) INPUT BEGIN DEGREES <CR>
6) INPUT END DEGREES <CR>, back to '1'

OE — Object Ellipse: See OC. Note: Major and minor radii must lie on centerline with one axis. Draws only right angle ellipses.

1) CURSOR TO ELLIPSE CENTER <CR>
2) OPTION: <CR> exits
3) CURSOR TO MINOR RADIUS <CR>
4) CURSOR TO MAJOR RADIUS <CR>
5) OPTION: Enter F for full ellipse
6) INPUT BEGIN DEGREES <CR>
7) INPUT END DEGREES <CR>, back to '1'

AW — Arrow: Draws arrows where indicated and in any radial orientation. Arrow sizes are constant on the world system.

1) OPTION: <CR> exits
2) CURSOR TO ARROW POINT <CR>
3) CURSOR TO ANY POINT ON ARROW SHAFT <CR>, back to '1'

GP — Get and Put: Gets a rectangular section of screen data and copies it to another place on the screen. Restricted in size by the GPN(n) array. Will not work without 256K RAM. "Save" GP data before exiting screen.

1) CURSOR TO TOP LEFT CORNER <CR>
2) OPTION: <CR> exits
3) CURSOR TO BOTTOM LEFT CORNER <CR>
4) CURSOR TO PUT POINT ON SCREEN <CR>, back to '1'

TX — Text: Prints strings where requested. Text size is constant on world system.

1) OPTION: <CR> exits
2) CURSOR TO TOP LEFT CORNER <CR>
3) INPUT TEXT <CR>, back to '1'

MS — Measure: First lists the world system position to the screen in X, Y coordinates, then lists the world system distance from that point to any other point on the screen in linear units (based on system units input on file from initial setup). Leaves a cross on the screen for reference. Not recorded.

1) CURSOR TO ANY POSITION <CR>, lists world XY
2) OPTION: <CR> exits and leaves cross
3) CURSOR TO ANY POSITION <CR>, lists the distance from original point. Back to '2'

Ron Smith is employed at the Lawrence Livermore National Laboratory as mechanical designer, and has had more than 20 years of experience as a draftsman.

The listing:

```
10 '*************************************************************************
20 '                                                                        *
30 ' TINYCAD, VERSION 1.1 -- 3/1/86 -- FOR SANYO MBC 5XX -- BY RON SMITH    *
40 '                                                                        *
50 '*************************************************************************
60 '
70 '
100   CLEAR
110 '*************************************************************************
120 '                                        DEFINE AND DIMENSION VARIABLES *
130 '*************************************************************************
140   DEFDBL F,S,X,Y:                                    'DOUBLE PRECISION
150   DEF FNSR=(X2-X1)/640:                                 'SCREEN RATIO
160   DEF FNY2=Y1+(.3125*(X2-X1)):                    'Y2 POSITION ON SCREEN
280 '
290 '
300   OCCOL=7:                                        'OBJECT CIRCLE COLOR
310   TXCOL=2:                                            'TEXT COLOR
320   OLCOL=7:                                       'OBJECT LINE COLOR
330   LLCOL=1:                                       'LEADER LINE COLOR
340   AWCOL=7:                                            'ARROW COLOR
345   CLCOL=2:                                            'ARROW COLOR
350   OECOL=7:                                       'OBJECT ELIPSE COLOR
380 '
390 '
400   WX=32767:                                       'MAXIMUM WORLD SIZE
410   TXH=20:TXV=10:                          'WIDTH AND HEIGHT OF TEXT STRINGS
420   AMAX=150:                                       'MAXIMUM ARROW SIZE
480 '
490 '
500   DIM FILENAME$(10):                                  'HOLDS 10 FILES
510   DIM LX(500),LY(500),LINESTOP(500),COL(500):         'DRAWS 500 LINES
520   DIM CCX(100),CCY(100),CRAD(100),CIRCLESTART(100):
530   DIM CIRCLEEND(100):                              'DRAWS 100 CIRCLES
540   DIM GX1(100),GX2(100),GX3(100),GY1(100):
550   DIM GY2(100),GY3(100),GPN(3000):     'GET AND PUT 100 TIMES & 3000 MATRIX
560   DIM AX1(100), AY1(100),ANG(100):                 'DRAWS 100 ARROWS
570   DIM TX(100),TY(100),TX$(100):                   'PRINTS 100 STRINGS
580   DIM CEX(100),CEY(100),ELIPSESTART(100),ELIPSEEND(100):
590   DIM ERAD(100),RATIO(100):                        'DRAWS 100 ELLIPSES
700 '
710 '
720   CLS:PRINT"TINYCAD":PRINT"BY RON SMITH":PRINT"VERSION 1.1   3/1/86
730   FOR T=1 TO 3000:NEXT T
740 '*************************************************************************
750 '                                                                MAIN MENU *
760 '*************************************************************************
770   CLS:LOCATE 10,30:PRINT"*** MAIN MENU ***
780   LOCATE 13,25:PRINT"1-WORK EXISTING FILE
790   LOCATE 15,25:PRINT"2-OPEN NEW FILE
800   LOCATE 17,25:PRINT"3-UTILITIES
810   LOCATE 19,40:INPUT"1 THRU 3";Q1
870 '
880 '
890   IF INT(Q1)<1 OR INT(Q1)>3 THEN 770
900 '*************************************************************************
910 '                                                 SEARCHES FILE LIST *
920 '*************************************************************************
930   IF Q1=1 THEN CLS:LOCATE 10,25:INPUT"NAME OF EXISTING FILE";FILENAME$
940   IF Q1=2 THEN CLS:LOCATE 10,25:INPUT"NAME OF NEW FILE";FILENAME$
950   IF FILENAME$=""THEN 740
960   OPEN "I",#1,"FILELIST"
970   INPUT#1,SIZE
980   FOR Z=0 TO SIZE
990   INPUT#1,FILENAME$(Z)
1000  NEXT Z
1010  CLOSE
1020  FOR Z=0 TO SIZE
```

```
103Ø  IF FILENAME$=FILENAME$(Z) THEN 1Ø5Ø
104Ø  NEXT Z
105Ø  IF Q1=1 AND FILENAME$<>FILENAME$(Z) THEN CLS:LOCATE 1Ø,25:PRINT
      "FILE NOT FOUND":FOR T= 1 TO 2ØØØ:NEXT T:GOTO 74Ø
106Ø  IF Q1=2 AND FILENAME$ =FILENAME$(Z) THEN CLS:LOCATE 1Ø,25:PRINT
      "FILE NAME TAKEN":FOR T= 1 TO 2ØØØ:NEXT T:GOTO 74Ø
107Ø  IF Q1=1 THEN 131Ø
108Ø '*************************************************************************
109Ø '                                                          ADDS NEW FILE *
11ØØ '*************************************************************************
111Ø  SIZE=SIZE+1:FILENAME$(SIZE)=FILENAME$
112Ø  OPEN "O",#1,"FILELIST"
113Ø  PRINT#1,SIZE
114Ø  FOR Z= Ø TO SIZE
115Ø  PRINT#1,FILENAME$(Z)
116Ø  NEXT Z
117Ø  CLOSE
118Ø '*************************************************************************
119Ø '                                  SETS WINDOW COORDINATES FOR NEW FILE *
12ØØ '*************************************************************************
121Ø  CLS
122Ø  PRINT"THERE ARE";WX;" HORIZ. POSITIONS ACROSS THE WORLD. HOW MANY ARE
123Ø  PRINT:INPUT"USED IN ONE UNIT OF MEASUREMENT";UN
124Ø  OPEN "O",#1,FILENAME$+".UNC"
125Ø  WRITE#1,UN
126Ø  CLOSE
127Ø  GOTO 134Ø
128Ø '*************************************************************************
129Ø '                                  GETS WINDOW COORDINATES FOR OLD FILE *
13ØØ '*************************************************************************
131Ø  OPEN "I",#1,FILENAME$+".UNC"
132Ø  INPUT#1,UN
133Ø  CLOSE
134Ø '*************************************************************************
135Ø '                                                          SETS WINDOW *
136Ø '*************************************************************************
137Ø  XØ=32Ø:YØ=1ØØ:                              'CENTER OF SCREEN FOR CURSOR
138Ø  X1=Ø:Y1=Ø:X2=WX:                            'VARIABLES FOR WINDOW SIZE
139Ø  IF Q1=2 THEN GOSUB 5ØØØ:                            'SKIP IF NEW FILE
14ØØ  CLS
141Ø  WINDOW(X1,Y1)-(X2,FNY2):                             'DEFINES WINDOW
142Ø  IF Q1=2 THEN 15ØØ:                                  'SKIP IF NEW FILE
143Ø '*************************************************************************
144Ø '                                  DRAWS OLD FILE ON FULL WINDOW *
145Ø '*************************************************************************
146Ø  GOSUB 8ØØØ
147Ø '*************************************************************************
148Ø '                                  REDEFINES SCREEN RATIO AND WINDOW *
149Ø '*************************************************************************
15ØØ  SR=FNSR:                                          'TEMPORARY SCREEN RATIO
151Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO TOP LEFT CORNER <CR>"
152Ø  GCURSOR(XØ,YØ),(XX1,YY1)
153Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO RIGHT SIDE <CR>"
154Ø  GCURSOR(XX1,YY1),(XX2,YY2)
155Ø  IF XX1=XX2 AND YY1=YY2 THEN 171Ø
156Ø  LINE (XX1*FNSR+X1,YY1*FNSR+Y1)-(XX2*FNSR+X1,(YY1+(.3125*(XX2-XX1)))*FNSR+
      Y1),,B:                          'DRAWS SCREEN RECTANGLE AROUND CHOICE
157Ø  GCURSOR(XX2,YY2),(XX3,YY3)
158Ø  IF XX2=XX3 AND YY2=YY3 THEN 16ØØ
159Ø  XX1=XX3:YY1=YY3: GOTO 153Ø:                    'TRY NEW SCREEN RECTANGLE
16ØØ  X1=XX1*SR:Y1=YY1*SR:X2=XX2*SR
161Ø  WINDOW(X1,Y1)-(X2,FNY2):                                'NEW WINDOW
162Ø  CLS
163Ø  IF Q1=2 THEN 171Ø
164Ø '*************************************************************************
165Ø '                                  DRAWS OLD FILE ON NEW WINDOW *
166Ø '*************************************************************************
167Ø  FUNCTION$="RD":GOSUB 8ØØØ
168Ø '*************************************************************************
169Ø '                                                  FUNCTION SELECTION *
17ØØ '*************************************************************************
171Ø  GOSUB 12ØØØ
172Ø  IF FUNCTION$="OL"THEN GOSUB 2ØØØ:GOTO 171Ø:                 'OBJECT LINE
1725  IF FUNCTION$="LL"THEN GOSUB 2ØØØ:GOTO 171Ø:                'LEADER LINE
1728  IF FUNCTION$="CL"THEN GOSUB 2ØØØ:GOTO 171Ø:           'CONSTRUCTION LINE
173Ø  IF FUNCTION$="OC"THEN GOSUB 216Ø:GOTO 171Ø:              'OBJECT CIRCLE
174Ø  IF FUNCTION$="GP"THEN GOSUB 237Ø:GOSUB 518Ø:GOTO 171Ø:     'GET AND PUT
175Ø  IF FUNCTION$="AW"THEN GOSUB 251Ø:GOTO 171Ø:                    'ARROW
176Ø  IF FUNCTION$="TX"THEN GOSUB 271Ø:GOTO 171Ø:                    'TEXT
```

```
177Ø  IF FUNCTION$="MS"THEN GOSUB 1211Ø:                          'MEASURE
178Ø  IF FUNCTION$="ER"THEN GOSUB 1231Ø:GOTO 171Ø:              'ERASE LAST
179Ø  IF FUNCTION$="RD"THEN Q1=1:GOTO 134Ø:                       'REDRAW
18ØØ  IF FUNCTION$="OE" THEN GOSUB 282Ø:GOTO 171Ø:             'OBJECT ELLIPSE
195Ø '
196Ø '
197Ø  IF FUNCTION$="SAVE"THEN GOSUB 5ØØØ:GOTO 171Ø
198Ø  IF FUNCTION$="END"THEN END
199Ø  GOTO 171Ø
2ØØØ '*************************************************************************
2Ø1Ø '                                                  SUBROUTINE DRAWS LINES *
2Ø2Ø '*************************************************************************
2Ø3Ø  GOSUB 12Ø6Ø:PRINT"CURSOR BEGENNING OF LINE <CR>"
2Ø4Ø  GCURSOR (XØ,YØ),(LX1,LY1)
2Ø5Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO END OF LINE <CR>
2Ø6Ø  GCURSOR (LX1,LY1),(LX2,LY2)
2Ø7Ø  COL(LN)=OLCOL
2Ø8Ø  IF FUNCTION$="LL"THEN COL(LN)=LLCOL
2Ø82  IF FUNCTION$<>"CL" THEN 2Ø9Ø
2Ø84  CLX1=FNSR*LX1+X1:CLY1=LY1*FNSR+Y1:CLX2=LX2*FNSR+X1:CLY2=LY2*FNSR+Y1
2Ø86  LINE(CLX1,CLY1)-(CLX2,CLY2),COL(LN)
2Ø87  IF CLX1=CLX2 AND CLY1= CLY2 THEN RETURN
2Ø88  GOTO 213Ø
2Ø9Ø  LX(LN)=FNSR*LX1+X1:LY(LN)=LY1*FNSR+Y1:LX(LN+1)=LX2*FNSR+X1:LY(LN+1)=LY2*
      FNSR+Y1
21ØØ  LINE(LX(LN),LY(LN))-(LX(LN+1),LY(LN+1)),COL(LN)
211Ø  LN=LN+1
212Ø  IF LX(LN-1)=LX(LN) AND LY(LN-1)=LY(LN) THEN LINESTOP(LN)=1:RETURN
213Ø  XØ=LX2:YØ=LY2
214Ø  LX1=LX2:LY1=LY2
215Ø  GOTO 2Ø5Ø
216Ø '*************************************************************************
217Ø '                                                SUBROUTINE DRAWS CIRCLES *
218Ø '*************************************************************************
219Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO CIRCLE CENTER <CR>
22ØØ  GCURSOR (XØ,YØ),(CCX,CCY)
221Ø  CCX(CN)=FNSR*CCX+X1:CCY(CN)=CCY*FNSR+Y1
222Ø  LINE(CCX(CN)-(FNSR*5),CCY(CN))-(CCX(CN)+(FNSR*5),CCY(CN)),LLCOL
223Ø  LINE(CCX(CN),CCY(CN)+(FNSR*3))-(CCX(CN),CCY(CN)-(FNSR*3)),LLCOL
224Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO RADIUS <CR> OR STAY <CR> TO END"
225Ø  GCURSOR (CCX,CCY),(CRX,CRY)
226Ø  IF CCX=CRX AND CCY=CRY THEN RETURN
227Ø  GOSUB 12Ø6Ø:INPUT"BEGIN DEGREES (OR 'F' FOR FULL CIRCLE) <CD>";FUNCTION$
228Ø  IF FUNCTION$="F" THEN CIRCLESTART(CN)=Ø
229Ø  IF FUNCTION$="F" THEN CIRCLEEND(CN)=36Ø:GOTO 231Ø
23ØØ  CIRCLESTART(CN)=VAL(FUNCTION$):GOSUB 12Ø6Ø:INPUT"END DEGREES <CR>";
      FUNCTION$:CIRCLEEND(CN)=VAL(FUNCTION$)
231Ø  CRAD(CN)=SQR((CCX-CRX)^2+((CCY-CRY)/.5Ø8ØØØ1)^2)*FNSR
233Ø  CIRCLE(CCX(CN),CCY(CN)),CRAD(CN),CIRCLESTART(CN)/36Ø,CIRCLEEND(CN)/36Ø,,
      OCCOL
234Ø  XØ=CCX:YØ=CCY
235Ø  CN=CN+1
236Ø  GOTO 216Ø
237Ø '*************************************************************************
238Ø '                                                SUBROUTINE TO GET AND PUT *
239Ø '*************************************************************************
24ØØ  GOSUB 12Ø6Ø:PRINT"CURSOR TO TOP LEFT CORNER <CR>"
241Ø  GCURSOR (XØ,YØ),(GX1(GN),GY1(GN))
242Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO BOTTOM RIGHT <CR>"
243Ø  GCURSOR (GX1(GN),GY1(GN)),(GX2(GN),GY2(GN))
244Ø  IF GX1(GN)=GX2(GN) AND GY1(GN)=GY2(GN) THEN RETURN
245Ø  GCURSOR (GX1(GN),GY1(GN)),(GX3(GN),GY3(GN))
246Ø  GET(GX1(GN),GY1(GN))-(GX2(GN),GY2(GN)),GPN
247Ø  PUT(GX3(GN),GY3(GN)),GPN,OR
248Ø  XØ=GX3(GN):YØ=GY3(GN)
249Ø  GN=GN+1
25ØØ  GOTO 237Ø
251Ø '*************************************************************************
252Ø '                                              SUBROUTINE TO DRAW ARROW HEAD *
253Ø '*************************************************************************
254Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO ARROW POINT <CR>"
255Ø  GCURSOR(XØ,YØ),(AX1,AY1)
256Ø  IF XØ=AX1 AND YØ=AY1 THEN RETURN
257Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO ANY POINT BEHIND (AND ON) ARROW LINE <CR>"
258Ø  GCURSOR(AX1,AY1),(AX2,AY2)
259Ø  IF AX1<=AX2 AND AY1>AY2 THEN ANG(AN)=ATN((AX2-AX1)*.5/(AY1-AY2)):
      GOTO 263Ø
26ØØ  IF AX1<AX2 AND AY1<=AY2 THEN ANG(AN)=ATN((AY2-AY1)/.5/(AX2-AX1))+1.57Ø7:
```

```
    GOTO 263Ø
261Ø  IF AX1>=AX2 AND AY1<AY2 THEN ANG(AN)=ATN((AX1-AX2)*.5/(AY2-AY1))+3.14159:
    GOTO 263Ø
262Ø  IF AX1>AX2 AND AY1>=AY2 THEN ANG(AN)=ATN((AY1-AY2)/.5/(AX1-AX2))+4.7124:
    GOTO 263Ø
263Ø  AX1(AN)=AX1*FNSR+X1:AY1(AN)=AY1*FNSR+Y1
264Ø  LINE(AX1(AN),AY1(AN))-(AX1(AN)+(SIN(ANG(AN)-.2)*AMAX)/.5,AY1(AN)-(COS
    (ANG(AN)-.2)*AMAX)),AWCOL
265Ø  LINE-(AX1(AN)+(SIN(ANG(AN)+.2)*AMAX)/.5,AY1(AN)-(COS(ANG(AN)+.2)*AMAX)),
    AWCOL
266Ø  LINE-(AX1(AN),AY1(AN)),AWCOL
267Ø  AN=AN+1
268Ø  XØ=AX1:YØ=AY1
269Ø  GOTO 255Ø
271Ø '*************************************************************************
272Ø '                                        SUBROUTINE PRINTS TEXT TO SCREEN *
273Ø '*************************************************************************
274Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO UPPER LEFT <CR>
275Ø  GCURSOR(XØ,YØ),(TX,TY)
276Ø  IF XØ=TX AND YØ=TY THEN RETURN
277Ø  GOSUB 12Ø6Ø:INPUT"TYPE TEXT <CR>";TX$(TN)
278Ø  TX(TN)=TX*FNSR+X1:TY(TN)=TY*FNSR+Y1
279Ø  SYMBOL(TX(TN),TY(TN)),TX$(TN),TXH,TXV,TXCOL
28ØØ  TN=TN+1:XØ=TX:YØ=TY
281Ø  GOTO 271Ø
282Ø '*************************************************************************
283Ø '                                              SUBROUTINE TO DRAW ELIPSE *
284Ø '*************************************************************************
285Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO ELIPSE CENTER <CR>
286Ø  GCURSOR(XØ,YØ),(CEX,CEY)
287Ø  CEX(EN)=FNSR*CEX+X1:CEY(EN)=CEY*FNSR+Y1
288Ø  LINE(CEX(EN)+FNSR*5,CEY(EN))-(CEX(EN)-FNSR*5,CEY(EN)),LLCOL
289Ø  LINE(CEX(EN),CEY(EN)+FNSR*3)-(CEX(EN),CEY(EN)-FNSR*3),LLCOL
29ØØ  GOSUB 12Ø6Ø:PRINT"CURSOR TO MINOR RAD <CR> OR STAY <CR> TO END"
291Ø  GCURSOR(CEX,CEY),(MNX,MNY)
292Ø  IF CEX=MNX AND CEY=MNY THEN RETURN
293Ø  GOSUB 12Ø6Ø:PRINT"CURSOR TO MAJOR RAD <CR>"
294Ø  GCURSOR(CEX,CEY),(MJX,MJY)
295Ø  GOSUB 12Ø6Ø:INPUT"BEGIN DEGREES (OR 'F' FOR FULL ELIPSE) <CR>";FUNCTION$
296Ø  IF FUNCTION$="F" THEN ELIPSESTART(EN)=Ø
297Ø  IF FUNCTION$="F" THEN ELIPSEEND(EN)=36Ø:GOTO 299Ø
298Ø  ELIPSESTART(EN)=VAL(FUNCTION$):GOSUB 12Ø6Ø:INPUT"END DEGREES <CR>";
    FUNCTION$:ELIPSEEND(EN)=VAL(FUNCTION$)
299Ø  IF CEX=MNX THEN MINRAD=ABS(CEY-MNY) ELSE MINRAD=ABS(CEX-MNX)
3ØØØ  XØ=LX1:YØ=LY1
3Ø1Ø  IF CEX=MJX THEN MAXRAD=ABS(CEY-MJY) ELSE MAXRAD=ABS(CEX-MJX)
3Ø2Ø  IF ABS(CEX-MNX)<ABS(CEY-MJY) THEN VERTRAD=MAXRAD ELSE VERTRAD=MINRAD
3Ø3Ø  IF VERTRAD=MAXRAD THEN HORIZRAD=MINRAD ELSE HORIZRAD=MAXRAD
3Ø4Ø  RATIO(EN)=VERTRAD/HORIZRAD
3Ø5Ø  IF MAXRAD=HORIZRAD THEN ERAD(EN)=HORIZRAD ELSE ERAD(EN)=VERTRAD
3Ø6Ø  ERAD(EN)=ERAD(EN)*FNSR
3Ø7Ø  CIRCLE(CEX(EN),CEY(EN)),ERAD(EN),ELIPSESTART(EN)/36Ø,ELIPSEEND(EN)/36Ø,
    RATIO(EN),OECOL
3Ø8Ø  XØ=CEX:YØ=CEY:EN=EN+1
3Ø9Ø  GOTO 282Ø
498Ø '
499Ø '
5ØØØ '*************************************************************************
5Ø1Ø '                          SUBROUTINE WRITES LINE DATA TO DISK FROM ARRAY *
5Ø2Ø '*************************************************************************
5Ø3Ø  OPEN "O",#1,FILENAME$+".OLL"
5Ø4Ø  PRINT#1,LN
5Ø5Ø  FOR Z= Ø TO LN
5Ø6Ø  WRITE#1,LX(Z),LY(Z),LINESTOP(Z),COL(Z)
5Ø7Ø  NEXT Z
5Ø8Ø  CLOSE
5Ø9Ø '*************************************************************************
51ØØ '                        SUBROUTINE WRITES CIRCLE DATA TO DISK FROM ARRAY *
511Ø ' *************************************************************************
512Ø  OPEN "O",#1,FILENAME$+".OCC"
513Ø  PRINT#1,CN
514Ø  FOR Z=Ø TO CN-1
515Ø  WRITE#1,CCX(Z),CCY(Z),CRAD(Z),CIRCLESTART(Z),CIRCLEEND(Z)
516Ø  NEXT Z
517Ø  CLOSE
518Ø '*************************************************************************
519Ø '                        SUBROUTINE WRITES GET PUT DATA TO DISK FROM ARRAY *
52ØØ '*************************************************************************
```

```
521Ø  AX=X2-X1:AY=FNY2-Y1
522Ø  FOR Z= Ø TO GN-1
523Ø  GX1(Z)=(GX1(Z)/64Ø)*AX+X1:GX2(Z)=(GX2(Z)/64Ø)*AX+X1:GX3(Z)=(GX3(Z)/64Ø)*
      AX+X1
524Ø  GY1(Z)=(GY1(Z)/2ØØ)*AY+Y1:GY2(Z)=(GY2(Z)/2ØØ)*AY+Y1:GY3(Z)=(GY3(Z)/2ØØ)*
      AY+Y1
525Ø  NEXT Z
526Ø  OPEN "O",#1,FILENAME$+".GPC"
527Ø  PRINT#1,GN
528Ø  FOR Z=Ø TO GN-1
529Ø  WRITE#1,GX1(Z),GX2(Z),GX3(Z),GY1(Z),GY2(Z),GY3(Z)
53ØØ  NEXT Z
531Ø  CLOSE
532Ø  IF FUNCTION$="GP"THEN RETURN
533Ø '*************************************************************************
534Ø '                    SUBROUTINE WRITES ARROWHEAD DATA TO DISK FROM ARRAY *
535Ø '*************************************************************************
536Ø  OPEN "O",#1,FILENAME$+".AWC"
537Ø  PRINT#1,AN
538Ø  FOR Z=Ø TO AN-1
539Ø  WRITE#1,AX1(Z),AY1(Z),ANG(Z)
54ØØ  NEXT Z
541Ø  CLOSE
542Ø '*************************************************************************
543Ø '                          SUBROUTINE WRITES TEXT DATA TO DISK FROM ARRAY *
544Ø '*************************************************************************
545Ø  OPEN "O",#1,FILENAME$+".TXT"
546Ø  PRINT#1,TN
547Ø  FOR Z=Ø TO TN-1
548Ø  WRITE#1,TX(Z),TY(Z),TX$(Z)
549Ø  NEXT Z
55ØØ  CLOSE
551Ø '*************************************************************************
552Ø '                        SUBROUTINE WRITES ELIPSE DATA TO DISK FROM ARRAY *
553Ø '*************************************************************************
554Ø  OPEN "O",#1,FILENAME$+".OEC"
555Ø  PRINT#1,EN
556Ø  FOR Z= Ø TO EN-1
557Ø  WRITE#1,CEX(Z),CEY(Z),ERAD(Z),ELIPSESTART(Z),ELIPSEEND(Z),RATIO(Z)
558Ø  NEXT Z
559Ø  CLOSE
797Ø '
798Ø '
799Ø  RETURN
8ØØØ '*************************************************************************
8Ø1Ø '                          SUBROUTINE READS LINE DATA FROM DISK AND DRAWS *
8Ø2Ø '*************************************************************************
8Ø3Ø  IF FUNCTION$="RD" THEN 81ØØ
8Ø4Ø  OPEN "I",#1,FILENAME$+".OLL"
8Ø5Ø  INPUT#1,LN
8Ø6Ø  FOR Z= Ø TO  LN
8Ø7Ø  INPUT#1,LX(Z),LY(Z),LINESTOP(Z),COL(Z)
8Ø8Ø  NEXT Z
8Ø9Ø  CLOSE
81ØØ  FOR Z=Ø TO LN-1
811Ø  IF LINESTOP(Z+1)=1 THEN 813Ø
812Ø  LINE (LX(Z),LY(Z))-(LX(Z+1),LY(Z+1)),COL(Z)
813Ø  NEXT Z
814Ø '*************************************************************************
815Ø '                        SUBROUTINE READS CIRCLE DATA FROM DISK AND DRAWS *
816Ø '*************************************************************************
817Ø  IF FUNCTION$="RD" THEN 824Ø
818Ø  OPEN "I",#1,FILENAME$+".OCC"
819Ø  INPUT#1,CN
82ØØ  FOR Z=Ø TO CN-1
821Ø  INPUT#1,CCX(Z),CCY(Z),CRAD(Z),CIRCLESTART(Z),CIRCLEEND(Z)
822Ø  NEXT Z
823Ø  CLOSE
824Ø  FOR Z= Ø TO CN-1
825Ø  LINE(CCX(Z)-(FNSR*5),CCY(Z))-(CCX(Z)+(FNSR*5),CCY(Z)),LLCOL
826Ø  LINE(CCX(Z),CCY(Z)+(FNSR*3))-(CCX(Z),CCY(Z)-(FNSR*3)),LLCOL
827Ø  CIRCLE(CCX(Z),CCY(Z)),CRAD(Z),CIRCLESTART(Z)/36Ø,CIRCLEEND(Z)/36Ø,,
      OCCOL
828Ø  NEXT Z
829Ø '*************************************************************************
83ØØ '                        SUBROUTINE READS GET PUT DATA FROM DISK AND DRAWS *
831Ø '*************************************************************************
832Ø  AX=X2-X1:AY=FNY2-Y1
833Ø  OPEN "I",#1,FILENAME$+".GPC"
```

```
8340  INPUT#1,GN
8350  FOR Z= 0 TO GN-1
8360  INPUT#1,GX1( Z),GX2( Z),GX3( Z),GY1( Z),GY2( Z),GY3( Z)
8370  NEXT Z
8380  CLOSE
8390  FOR Z=0 TO GN-1
8400  GX1(Z)=((GX1(Z)-X1)/AX)*640:GX2(Z)=((GX2(Z)-X1)/AX)*640:GX3(Z)=((GX3(Z)-
      X1)/AX)*640
8410  GY1(Z)=((GY1(Z)-Y1)/AY)*200:GY2(Z)=((GY2(Z)-Y1)/AY)*200:GY3(Z)=((GY3(Z)-
      Y1)/AY)*200
8420  NEXT Z
8430  FOR Z=0 TO GN-1
8440  IF GX1(Z)<1 OR GX2(Z)>639 THEN 8500
8450  IF GX3(Z)<1 OR GX3(Z)>639 THEN 8500
8460  IF GY1(Z)<1 OR GY2(Z)>199 THEN 8500
8470  IF GY3(Z)<1 OR GY3(Z)>199 THEN 8500
8480  GET(GX1(Z),GY1(Z))-(GX2(Z),GY2(Z)),GPN
8490  PUT(GX3(Z),GY3(Z)),GPN,OR:          '"OR" PUTS OVER ALL, ERASES BACKGROUND
8500  NEXT Z
8510 '*************************************************************************
8520 '                  SUBROUTINE READS ARROWHEAD DATA FROM DISK AND DRAWS *
8530 '*************************************************************************
8540  IF FUNCTION$="RD" THEN 8610
8550  OPEN "I",#1,FILENAME$+".AWC"
8560  INPUT#1,AN
8570  FOR Z=0 TO AN-1
8580  INPUT#1,AX1(Z),AY1(Z),ANG(Z)
8590  NEXT Z
8600  CLOSE
8610  FOR Z=0 TO AN-1
8620  LINE(AX1(Z),AY1(Z))-(AX1(Z)+(SIN(ANG(Z)-.2)*AMAX)/.5,AY1(Z)-(COS(ANG(Z)-
      .2)*AMAX)),AWCOL
8630  LINE-(AX1(Z)+(SIN(ANG(Z)+.2)*AMAX)/.5,AY1(Z)-(COS(ANG(Z)+.2)*AMAX)),AWCOL
8640  LINE-(AX1(Z),AY1(Z)),AWCOL
8650  NEXT
8660  CLOSE
8670 '*************************************************************************
8680 '                     SUBROUTINE READS TEXT DATA FROM DISK AND PRINTS *
```

```
8690 '*************************************************************************
8700  IF FUNCTION$="RD" THEN 8770
8710  OPEN "I",#1,FILENAME$+".TXT"
8720  INPUT#1,TN
8730  FOR Z=0 TO TN-1
8740  INPUT#1,TX(Z),TY(Z),TX$(Z)
8750  NEXT Z
8760  CLOSE
8770  FOR Z=0 TO TN-1
8780  SYMBOL(TX(Z),TY(Z)),TX$(Z),TXH,TXV,TXCOL
8790  NEXT Z
8800 '*************************************************************************
8810 '                        SUBROUTINE READS ELIPSE DATA FROM DISK AND DRAWS *
8820 '*************************************************************************
8825 IF FUNCTION$="RD" THEN 8890
8830  OPEN "I",#1,FILENAME$+".OEC"
8840  INPUT#1,EN
8850  FOR Z= 0 TO EN-1
8860  INPUT#1,CEX(Z),CEY(Z),ERAD(Z),ELIPSESTART(Z),ELIPSEEND(Z),RATIO(Z)
8870  NEXT Z
8880  CLOSE
8890  FOR Z= 0 TO EN-1
8900  LINE(CEX(Z)-(FNSR*5),CEY(Z))-(CEX(Z)+(FNSR*5),CEY(Z)),LLCOL
8910  LINE(CEX(Z),CEY(Z)+(FNSR*3))-(CEX(Z),CEY(Z)-(FNSR*3)),LLCOL
8920  CIRCLE(CEX(Z),CEY(Z)),ERAD(Z),ELIPSESTART(Z)/360,ELIPSEEND(Z)/360,
      RATIO(Z),OECOL
8930  NEXT Z
11970 '
11980 '
11990 RETURN
12000 '*************************************************************************
12010 '                                        SUB ROUTINE FUNCTION PROMPT *
12020 '*************************************************************************
12030 LOCATE 24,1:PRINT"                                                      "
12040 LOCATE 24,1:INPUT "FUNCTION";FUNCTION$
12050 LOCATE 24,1:PRINT"                                                      ":
      RETURN
12060 '*************************************************************************
12070 '                                                SUB ROUTINE LOCATE *
12080 '*************************************************************************
12090 LOCATE 24,1:PRINT"                                                      "
12100 LOCATE 24,1:RETURN
12110 '*************************************************************************
12120 '                                        SUBROUTINE FOR MEASURING *
12130 '*************************************************************************
12140 GCURSOR(X0,Y0),(XX,YY)
12150 XZ=XX:YZ=YY
12160 XX1=FNSR*XX+X1:YY1=FNSR*YY+Y1
12170 GOSUB 12060
12180 PRINT USING"#####.##";XX1/UN,YY1/UN
12190 GCURSOR(XX,YY ),(XXX,YYY)
12200 X0=XXX:Y0=YYY
12210 IF XX=XXX AND YY=YYY THEN 12270
12220 DIS=SQR((XZ-XXX)^2+((YZ-YYY)/.508001)^2)
12230 GOSUB 12060
12240 PRINT USING"#####.##";FNSR*DIS/UN
12250 XX=XXX:YY=YYY
12260 GOTO 12190
12270 XX=FNSR*XX+X1:YY=YY*FNSR+Y1
12280 LINE(XX-(FNSR*5),YY)-(XX+(FNSR*5),YY),OLCOL
12290 LINE(XX ,YY +(FNSR*3))-(XX ,YY -(FNSR*3)),OLCOL
12300 RETURN
12310 '*************************************************************************
12320 '                                                ERASES LAST FUNCTION *
12330 '*************************************************************************
12340 GOSUB 12060
12350 INPUT"FUNCTION TO ERASE LAST OR 'X' TO CANCEL <CR>";Q3$
12360 IF Q3$="X"THEN RETURN
12370 IF Q3$="OL"OR Q3$="LL"THEN LINESTOP(LN)=0:LN=LN-1:LINESTOP(LN)=1:
      LX(LN)=LX(LN-1):LY(LN)=LY(LN-1)
12380 IF Q3$="OC"THEN CN=CN-1
12390 IF Q3$="GP"THEN GN=GN-1
12400 IF Q3$="OE"THEN EN=EN-1
12410 IF Q3$="AW"THEN AN=AN-1
12420 IF Q3$="TX"THEN TN=TN-1
12430 RETURN
12440 CLOSE
```

ASK SANYO

The following questions have been answered by personnel at Sanyo Business Systems Corp. as a service to SOFT SECTOR readers. Should you have a question, mail it to: 51 Joseph Street, Moonachie, NJ 07074.

Q. *I am a graduate student studying paleontology. In my research I work with small fossils that require many tedious measurements for identification purposes.*

I would like to set up a system by which I could place a specimen under a video camera attached to a microscope. The image would then be fed into a monitor. Using a mouse, joystick or preferably a plotting board with a puck, I could then point out on the screen the measurements I wish to make, which would be stored in a database for later analysis. I know that this can be done with an IBM PC system. I would like to know if it could be done with my Sanyo MBC-555. Do you know of any fundamental limitations in the 555's graphics capabilities that would make this process infeasible?

Steve Hageman
Urbana, IL

A. There is no limitation on the MBC-555's graphics that I know of. In fact, in most respects they are superior to that of the IBM PC (we have 640 by 200 pixel resolution with eight colors). Interfacing certain products can be difficult due to the lack of an accessible bus.

I have written test programs for the MBC-555 that use a mouse and interface with BASIC (Sanyo's version for the MBC-555). The program is actually a poor man's (actually a penniless man's) *PC-Paint*, but it will show you how to interface assembly language programs with a high level language and how to use RS-232 interrupts. However, we cannot and will not support the program in any manner whatsoever.

If you are interested in obtaining source listings of the program, send a formatted disk and $15 for postage and handling to us and I will supply you with the code. You will need the *Macro Assembler* to change anything. Remember, once you get the code, you are on your own.

Q. *I am writing in the hope that you can tell me how to "cure" two problems I have with my Sanyo 550.*

One problem is with the `VAL(n)` *function in Sanyo* BASIC *(I have version 1.31). I seem to get a syntax error a large amount of the time when I use this function. Could this be a malfunction of my hardware, my software, or is there something I should be doing and am not?*

The other major problem I have been having deals with MS-DOS Version 2.11. I typed in the print spooler from the January '86 SOFT SECTOR*, Page 51, and set up the* `CONFIG.SYS` *file (which I would have needed anyway). This worked fine. The problem came when I read Danny Humphress' "Mastering MS-DOS" column. He said that you should add the line* `DEVICE=ANSI.SYS` *to your* `CONFIG.SYS` *file. OK, there is no* `ANSI.SYS` *file on the 550 master disk. I discussed this with my local Sanyo dealer, and he offered to copy the* `ANSI55.SYS` *file for me. When I got home, I put this on a copy of my system disk and added* `ANSI55.SYS` *to my* `CONFIG.SYS` *file. When I tried to reboot my system, I got the normal MS-DOS messages. However, the time and date prompts did not display properly. Could you tell me how to patch and/or correct this problem?*

Mrs. Mickey Raymer
Westville, FL

A. There is a problem with the VAL statement in releases before BASIC 1.35. The problem is fixed (along with a problem in the joystick command) in BASIC 1.35 which comes with the MBC-555 Systems Utilities Disk. This disk may be purchased from Sanyo for $15. This fix is for Sanyo BASIC only. It does not apply to the version of GW-BASIC that comes with the video board.

If you have already purchased the utilities disk and did not receive version 1.35, you may have your disk updated by sending Sanyo the *original* utilities disk along with $2.50 for postage and handling.

`ANSI55.SYS` is a device driver for the video board version of MS-DOS only. It will not function correctly with the regular Sanyo video output. In fact, I'm surprised that it worked as well as you described since `ANSI55.SYS` is hardware dependent.

Anyhow, one way around the problem might be to put a file called ANSI.SYS on the disk. Many programs look for this file without really checking to see if it is really a device driver and has been loaded into the system. You can take any file and name it ANSI.SYS, but don't try to load it using CONFIG.SYS. I have found this to be a solution in some, but not all, cases.

Q. *I've recently been looking into hard disk systems for the Sanyo 550 and have noticed that they require one to boot the system from a floppy. This seems to be a waste of hard disk use. Most other manufacturers boot from a hard disk, once it is formatted and the system loaded.*

Unless I am missing something, why can't the hard disk be configured from the floppy, then appropriate internal wiring modified to direct the hard disk as the default?

Rodney Starcher
Akron, OH

A. First, I assume you have an MBC-555 that you are using with a hard disk. The PC compatibles (of which the MBC-555 is not one) have controllers containing a hard disk BIOS in ROM. When a PC is booted, the PC BIOS looks for other resident BIOS starting at 0C000 Hex and passes control to any that it finds. Since the hard disk BIOS is normally located at 0C8000 Hex, it takes control and starts the boot procedure from the fixed drive if a floppy is not in the 'A' drive.

This procedure cannot be followed on the MBC-555 because the entire BIOS is located on disk (except for a small bootstrap program in ROM). Therefore, the MBC-555 must be booted from a floppy just to load the BIOS and MS-DOS. From there, a device driver is loaded that handles the fixed disk.

The amount of wiring and programming that would have to be done to enable the system to boot from a hard disk would be more than enormous. In fact, you would probably be designing a whole new computer.

Q. *What I wish to do is to have GW-BASIC stop an entered input after a certain number of characters have been typed by the user. Sanyo BASIC accomplishes this with* INPUT(i)*; is there a way to do this with GW-BASIC?*

George E. Banks, Jr.
Jacksonville, FL

A. GW-BASIC has a command similar to the Sanyo BASIC INPUT(i), <var>. Its syntax is X$ = INPUT$(i) and it is only good for string input. Also, it does not echo the characters as they are typed. The function is terminated as soon as 'i' characters are typed whereas the Sanyo routine needs an explicit <cr> to end the input.

Q. *We make use of a legal database to conduct research into jurisprudence, and use* Intellicom *as our communications package. The problem we face is that text downloaded from the database is not compatible with* WordStar. *We would like to be able to edit out extraneous cases and system instructions. This would allow us to incorporate this data directly into court briefs. The stored files all seem to contain strings of extra characters (usually ^@) in such a number that they cannot easily be removed. Do you have any suggestions as to how to capture* WordStar *compatible files?*

Eric R. Hutton
Bridgewater Legal Centre
Bridgewater, Nova Scotia

A. The character you are seeing is the ASCII NULL character which is probably being sent by the mainframe. Here is a small BASIC program which will filter out those characters.

```
100 INPUT "Enter name of input file:
    ", IFILE$
110 INPUT "Enter name of output
    file: ", OFILE$
120 OPEN "r", #1, IFILE$,1
130 OPEN "r", #2, OFILE$,1
140 FIELD 1, 1 AS CHARIN$
150 FIELD 2, 1 AS CHAROUT$
160 SIZE = LOF(1) : PRINT "File size
    is "; SIZE$ : PRINT
170 FOR I = 1 TO SIZE
180 GET 1
190 IF LEFT$(CHARIN$,1) = CHR$(0)
    GOTO 230
200 RSET CHAROUT$ = CHARIN$
210 PUT 2
220 PRINT CHAROUT$;
230 NEXT I
240 CLOSE 1
250 CLOSE 2
260 END
```

If the ^@ is not actually the NULL character that I suspect it is, the only line that must be changed is 190. Instead of CHR$(0), put in the ASCII character you wish to filter. For example, if you want to filter out all occurrences of lowercase 'a', change Line 190 to:

```
190 IF LEFT$(CHARIN$,1) = "a"
  GOTO 230
```

or

```
190 IF LEFT$(CHARIN$,1) = CHR$(97)
  GOTO 230
```

Keep Track of Your Stock Market Bulls and Bears

By Robert J. Craig

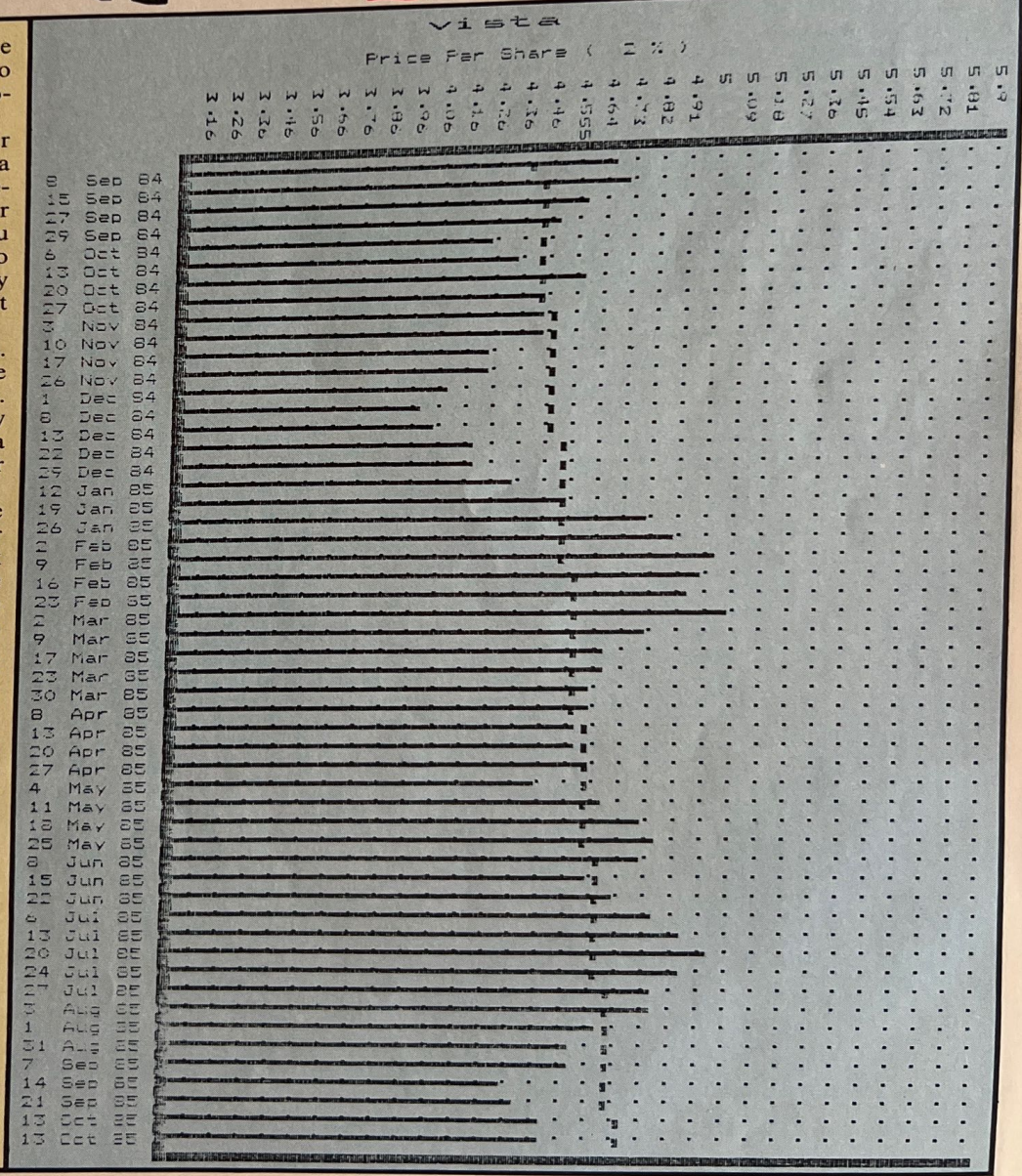
For those who have just entered the stock market and found it tough to follow investments, here is a program to make the job easier.

Stock Tracker keeps track of your investments. Through weekly input, a record can be kept of your stock's price-per-share trend, its current value, your current profit (or loss) and how much you have invested. The program is set up to track 10 stocks. If you have more, copy the program to a separate disk and use it for the additional stocks.

Stock Tracker has three main parts. Stock Trend Input allows you to enter the current price-per-share for all your stocks. It displays a chart that shows the weekly price trend of your stock (provided data is entered once a week) and the number of shares owned over the past year.

A break-even line is plotted to show the current price-per-share at which your invested amount just equals your stock value. If the stock trend is above the line you are in the black, but if it's below the line you are in the red. Several people advise that it is a good time to buy when you are below this line since it lowers your dollar cost average. This is true only if the stock is expected to rise in the future.

If you'd like, a hard copy of the price trend graph, which covers the entire data set for the stock, can be printed. A regression line is drawn on the graph showing the averaged trend of the stock. The printer codes are set up for a Gemini-10X and the program uses the following features: downloadable characters, adjusta-



Robert Craig, a captain in the U.S. Air Force, is working on his master's in atmospheric science. He may be contacted at 1225 W. Prospect Road, F204, Fort Collins, CO 80526, phone (303) 224-4079. Please enclose an SASE for a reply when writing.

ble vertical motion index, double width, enhanced or bold print and printer reset. These are the codes you'll need to change if you have a different printer. Each code sequence is commented to make this easier.

The second part of the program, Stock Value Display, displays on a simulated three-dimensional bar graph your stocks' current value, the amount you've invested, and your profits or losses. You have the option of changing the scale to get the most meaningful presentation. The scale options are 0-$800, 0-$8,000 and 0-$80,000.

Also displayed in digital format is the total amount invested and your total profit or loss. The stock names displayed under each bar graph contain the first three letters of the true name plus the last letter of the name. This was done since many stock names start with the same letters.

The third part of the program, Stock Transaction Input, allows you to enter stock purchases, sales and dividends. All files are updated for each transaction. There is also an option to print out a record of all your transactions for each stock.

Every stock has three separate files of data. The files with the extension .1ST hold all the trend data. Files with .STK hold the current amount invested and number of shares. Files with .PUR hold the record of all your stock transactions. One other file holds the stock names (STKNAMES.DAT). All these files are initially created by the program when a new stock is opened.

Stock Tracker was written for a color screen and might need to be modified to look nice on a monochrome monitor. BASIC Version 1.32 and MS-DOS 2.11 were also used. Since this program is data dependent, the display charts won't show much until a database has been built up.

Since I'm an amateur in the stock market I probably left out some features that could be useful. If you think of anything to enhance this program, let me know. Good luck and feel free to call if you have problems with the program or changing printer codes.

The listing:

```
1Ø REM PROGRAM STOCK
2Ø REM By Robert J. Craig, 1225 W. Prspect Rd #f2Ø4, Fort Collins, CO 8Ø526
3Ø REM 1-3Ø3-224-4Ø79
4Ø REM ************************************************************
5Ø REM * This program stores stock data and displays it in several  *
6Ø REM * modes.  Input data consists of price-per-share data, stock *
7Ø REM * purchases and sales.  Data is output in 3-D similated bar  *
75 REM * graph (stock values and amount invested), a line graph (   *
8Ø REM * stock price-per-share record for year ), and a printed bar *
85 REM * graph (stock price-per-share continuous record).  The      *
87 REM * printer codes are set up for a Gemini-1ØX.                 *
9Ø REM ************************************************************
1ØØ REM ******************************
1Ø5 REM * DRAW INTRODUCTION SCREEN *
11Ø REM ******************************
115 COLOR 2,Ø
12Ø CLS:SYMBOL(226,2),"STOCK",5,4,6,Ø:SYMBOL(24Ø,45),"Tracker",3,2,4,Ø
125 LOCATE 12,33,Ø:COLOR 2:PRINT "By Robert Craig":LOCATE 13,27:PRINT "1225 W. P
rospect Rd, #f2Ø4"
13Ø LOCATE 14,29:PRINT"Fort Collins, 8Ø526"
135 LOCATE 15,33:PRINT"1-3Ø3-224-4Ø79"
14Ø LOCATE 24,3Ø:COLOR 3:PRINT "Loading data arrays";
16Ø DIM STOCK(3,2Ø8),ROTNUM(1Ø,14),MONLEN(12),MON$(12),LPPPS(3Ø)
17Ø GOSUB 134ØØ
18Ø LOCATE 24,27:COLOR 4:PRINT SPACE$(55);TAB(27);"Press any key to continue";
185 IF INKEY$="" THEN 185
2ØØ REM *************
2Ø5 REM * MAIN MENU *
21Ø REM *************
215 CLS:LOCATE 9,25:COLOR 6,Ø:PRINT "Push Number Of Selection"
22Ø LOCATE 12,3Ø:COLOR 2:PRINT"1. Input weekly stock data"
225 LOCATE 14,3Ø:PRINT"2. Display current stock information"
23Ø LOCATE 16,3Ø:PRINT"3. Input stock purchases/sales"
232 LOCATE 18,3Ø:PRINT"4. Open new stock file"
234 LOCATE 2Ø,3Ø:PRINT"5. Delete stock file"
235 LOCATE 22,3Ø:PRINT"6. Return to operating system"
24Ø CHOICE=VAL(INKEY$):IF CHOICE=Ø THEN 24Ø ELSE 245
245 IF CHOICE<7 THEN LOCATE CHOICE*2+1Ø,29:COLOR ,7:PRINT CHOICE
25Ø COLOR ,Ø:FOR A=1 TO 5ØØ:NEXT A
255 ON CHOICE GOTO 3ØØ,4ØØ,7ØØ,9ØØ,11ØØ
27Ø IF CHOICE=6 THEN SYSTEM
275 GOTO 2ØØ
29Ø REM ************************
292 REM * STOCK TREND DISPLAY *
294 REM ************************
3ØØ CLS:COLOR 7,Ø:LOCATE 1,28,Ø:PRINT "STOCK INFORMATION DISPLAY"
3Ø5 LOCATE 5,1:COLOR 2:PRINT "Enter price-per-share for each stock shown."
31Ø LOCATE 6,1:PRINT "If PPS is not available, enter Ø."
312 GOSUB 1239Ø
315 GOSUB 11435
32Ø REM ********** Update stock files for each stock
325 LOCATE 24,1:COLOR 3:PRINT SPACE$(55);TAB(1);"Updating stock files, stand by.
"
```

```
335 FOR A=Ø TO STKNUM
34Ø    SNAME$=NAMES$(A)
342    PPS=PPS(A)
345    IF PPS(A)<>Ø THEN GOSUB 1Ø215:GOSUB 1Ø125
355 NEXT A
36Ø REM ********** Display Data
365 FOR C=Ø TO STKNUM
37Ø    SNAME$=NAMES$(C)
375    PPS=PPS(C)
38Ø    GOSUB 1Ø215:GOSUB 1ØØ9Ø:GOSUB 1ØØØØ
385 NEXT C
39Ø GOTO 2ØØ
4ØØ REM ************************
4Ø5 REM * STOCK VALUE DISPLAY *
41Ø REM ************************
411 CLS:COLOR 7,Ø:LOCATE 1,3Ø:PRINT "STOCK VALUE DISPLAY"
415 GOSUB 1239Ø
43Ø LOCATE 5,1:COLOR 2:PRINT "Enter price-per-share for each stock shown."
435 LOCATE 6,1:PRINT "If PPS is not available, enter Ø."
44Ø GOSUB 11435
445 REM ********** Load in stock data
45Ø TOTALCOST=Ø:TOTALVALUE=Ø
455 FOR A=Ø TO STKNUM
46Ø    OPEN "i",2,NAMES$(A)+".stk"
465       INPUT#2,COST(A):INPUT#2,SHARES(A)
47Ø       TOTALCOST=TOTALCOST+COST(A)
475       TOTALVALUE=TOTALVALUE+PPS(A)*SHARES(A)
48Ø    CLOSE 2
485 NEXT A
49Ø PROFITS=TOTALVALUE-TOTALCOST
495 REM ********** Display stock information
5ØØ FIRSTPASS=1
51Ø GOSUB 132ØØ:GOSUB 133ØØ
515 HEADING$="STOCK VALUES AS OF "+MON$+" "+DAY$+" 19"+RIGHT$(DATE$,2)
52Ø GOSUB 12ØØØ
53Ø FOR A=Ø TO STKNUM
535    BAR=82+A*42
545    VALUE=COST(A):CB=3:CL=5:GOSUB 12175
55Ø    BAR=94+A*42
555    VALUE=PPS(A)*SHARES(A):CB=2:CL=4:GOSUB 12175
565    LABEL$=LEFT$(NAMES$(A),3)+RIGHT$(NAMES$(A),1):GOSUB 12335
57Ø NEXT A
575 BAR=5Ø3:VALUE=TOTALCOST:CB=3:CL=5:GOSUB 12175
58Ø BAR=515:VALUE=TOTALVALUE:CB=2:CL=4:GOSUB 12175
585 LABEL$="TOTL":GOSUB 12335
59Ø BAR=546:VALUE=ABS(PROFITS)
595 IF PROFITS>Ø THEN CB=3:CL=4 ELSE CB=4:CL=5
6ØØ GOSUB 12175:LABEL$=" PRFT":GOSUB 12335
6Ø2 LOCATE 25,41:COLOR 7:PRINT "Spent-          Profit-          ";
6Ø4 LOCATE 25,47:COLOR 1,6:PRINT USING "######.##";TOTALCOST;TAB(65);PROFITS;
6Ø5 A$="Ø"
61Ø LOCATE 25,8,Ø:COLOR 4,Ø:PRINT "Options:";
615 WHILE VAL(A$)<1 OR VAL(A$)>4
62Ø    LOCATE 25,2Ø,Ø:COLOR 3:PRINT "1-Scale(Ø-8ØØ)  ";:GOSUB 125Ø5
625    LOCATE 25,2Ø:PRINT "2-Scale(Ø-8ØØØ) ";:GOSUB 125Ø5
63Ø    LOCATE 25,2Ø:PRINT "3-Scale(Ø-8ØØØØ)";:GOSUB 125Ø5
635    LOCATE 25,2Ø:COLOR 4:PRINT "4-Menu              ";:GOSUB 125Ø5
64Ø WEND
645 IF A$="4" THEN GOTO 2ØØ
65Ø ON VAL(A$) GOSUB 1243Ø,1248Ø,12455
655 A$="Ø":GOTO 615
7ØØ REM ********************************
7Ø5 REM * INPUT STOCK PURCHASE OR SALE *
71Ø REM ********************************
715 CLS:COLOR 2,Ø:LOCATE 1,34,Ø:COLOR 7:PRINT "STOCK UPDATE"
717 GOSUB 1239Ø
72Ø ORIGDATE$=DATE$
725 LOCATE 4,1:INPUT "Enter date of transaction(mm dd yy). ",A$
727 IF VAL(LEFT$(A$,2))>12 THEN 715
73Ø DATE$=A$:LOCATE 4,1:PRINT SPACE$(6Ø)
735 STOCKAMOUNT=Ø:NEWSHARES=Ø
74Ø LOCATE 5,1:COLOR 2:INPUT (1) "Is this a stock purchase or sale,p/s? ",A$
745 IF A$="s" THEN GOSUB 13125
75Ø LOCATE 5,1:PRINT SPACE$(5Ø);TAB(1);"Enter number of shares purchased."
755 LOCATE 6,1:PRINT "If purchase is dividends, begin number with d. "
76Ø LOCATE 5,35:COLOR 7:INPUT NEWSHARES$
765 IF LEFT$(NEWSHARES$,1)="d" THEN GOSUB 13Ø75 ELSE GOSUB 13Ø95
77Ø REM *****DETERMINE WHICH STOCK IS INVOLVED
775 LOCATE 8,1:PRINT "Push number of stock involved."
```

```
78Ø FOR A=Ø TO STKNUM
785    LOCATE 1Ø+A,5:COLOR 2:PRINT USING "## &        &";A;NAMES$(A)
79Ø NEXT A
795 B$=INKEY$:IF B$="" THEN 795
8ØØ IF VAL(B$)=<STKNUM THEN LOCATE 1Ø+VAL(B$),5:COLOR Ø,2:PRINT VAL(B$) ELSE 795
8Ø5 REM ***** INPUT STOCK FILES
81Ø LOCATE 8,1:COLOR 2,Ø:PRINT SPACE$(55)
815 LOCATE 23,1:INPUT "Save new stock transaction,y/n? ",A$
82Ø IF A$<>"y" THEN 87Ø
825 OPEN "i",2,NAMES$(VAL(B$))+".stk"
83Ø    INPUT#2, PREVSTOCKAMOUNT:INPUT#2, PREVSHARES
831    STOCKC=STOCKAMOUNT:SHARESC=NEWSHARES
835    STOCKAMOUNT=STOCKAMOUNT+PREVSTOCKAMOUNT
84Ø    NEWSHARES=NEWSHARES+PREVSHARES
845    OPEN "o",3,NAMES$(VAL(B$))+".stk"
85Ø        PRINT#3,USING "#####.##";STOCKAMOUNT:PRINT#3,USING "####.###";NEWSHAR
ES
855    CLOSE 3
86Ø CLOSE 2
865 GOSUB 13ØØØ:REM update stock trend list
87Ø REM ***** RESET FOR MORE STOCK INPUTS
875 LOCATE 23,1:COLOR 2,Ø:PRINT SPACE$(7Ø);TAB(1);:INPUT "Anymore stock purchase
s to enter,y/n? ",A$
88Ø IF A$="y" THEN 7ØØ
882 LOCATE 23,1:PRINT SPACE$(7Ø);TAB(1);:INPUT "Would you like a record of purch
ases/sales,y/n? ",A$
883 IF A$="y" THEN GOSUB 135ØØ
885 DATE$=ORIGDATE$
89Ø GOTO 2ØØ
9ØØ REM ************************
9Ø5 REM * OPEN NEW STOCK FILE *
91Ø REM ************************
915 CLS:LOCATE 1,3Ø,Ø:COLOR 7:PRINT"START NEW STOCK FILES"
917 ON ERROR GOTO 14ØØØ
92Ø GOSUB 1239Ø
925 IF STKNUM<9 THEN GOTO 95Ø
93Ø LOCATE 1Ø,1:COLOR 4:PRINT "With this stock, you have exceeded the max number
 of stocks the program can     handle.  To track more stocks, copy program on to
```

```
 new disk and then enter new   stocks."
935 LOCATE 15,1:COLOR 1:PRINT "Press any key to return to menu."
94Ø A$=INKEY$:IF A$="" THEN 94Ø
945 GOTO 2ØØ
95Ø LOCATE 5,1:COLOR 2:PRINT "Enter name of stock.  Name must be eight character
s or less and it must be      unique from other stocks stored.";
955 INPUT (8) STKNAME$
1ØØØ REM *****Create .stk file
1ØØ5 OPEN "o",2,STKNAME$+".stk"
1Ø1Ø    PRINT#2,Ø:PRINT#2,Ø
1Ø15 CLOSE 2
1Ø2Ø REM ***** Add name to name file
1Ø25 OPEN "o",2,"stknames.dat"
1Ø3Ø    PRINT#2,STKNUM+1
1Ø35    FOR A=Ø TO STKNUM
1Ø4Ø       PRINT#2,NAMES$(A)
1Ø45    NEXT A
1Ø5Ø    PRINT#2,STKNAME$
1Ø55 CLOSE 2
1Ø6Ø LOCATE 2Ø,1:COLOR 3:PRINT "New file created under name ";STKNAME$;"."
1Ø65 FOR A=1 TO 15ØØ:NEXT A
1Ø7Ø GOTO 2ØØ
11ØØ REM **********************
11Ø5 REM * DELETE STOCK *
111Ø REM **********************
1115 CLS:COLOR 7,Ø:LOCATE 1,32,Ø:PRINT "DELETE STOCK FILE"
1117 GOSUB 1239Ø
112Ø LOCATE 5,1:COLOR 2:PRINT "Push number of file to delete"
1125 FOR A=Ø TO STKNUM
113Ø    LOCATE 1Ø+A,5:COLOR 2:PRINT USING "## &      &";A;NAMES$(A)
1135 NEXT A
114Ø B$=INKEY$:IF B$="" THEN 114Ø
1145 IF VAL(B$)=<STKNUM THEN LOCATE 1Ø+VAL(B$),5:COLOR Ø,2:PRINT VAL(B$) ELSE 11
4Ø
1147 DELNAME$=NAMES$(VAL(B$))
115Ø FOR B=VAL(B$)+1 TO STKNUM
1155    NAMES$(B-1)=NAMES$(B)
116Ø NEXT B
```

```
1165 OPEN "o",2,"stknames.dat"
117Ø    PRINT#2,STKNUM-1
1175    FOR C=Ø TO STKNUM-1
118Ø       PRINT#2,NAMES$(C)
1185    NEXT C
119Ø CLOSE 2
1195 LOCATE 2Ø,1:COLOR 3,Ø:PRINT "File ";DELNAME$;" has been deleted."
12ØØ FOR A=1 TO 15ØØ:NEXT A
12Ø5 GOTO 2ØØ
1ØØØØ REM ---------- Display stock trend data ----------
1ØØØ5 MIN=STOCK(2,CT):MAX=MIN:COUNT=CT
1ØØ1Ø FOR A=BT TO 52
1ØØ2Ø    IF STOCK(2,COUNT)>=MAX THEN MAX=STOCK(2,COUNT)
1ØØ25    IF STOCK(2,COUNT)<MIN THEN MIN=STOCK(2,COUNT)
1ØØ27    COUNT=COUNT+1
1ØØ3Ø NEXT A
1ØØ35 GOSUB 1Ø935
1ØØ4Ø GOSUB 11285:' Plot line and display date
1ØØ45 ' Wait for option
1ØØ5Ø CHOICE=VAL(INKEY$):IF CHOICE=Ø THEN 1ØØ5Ø
1ØØ55 IF CHOICE=1 THEN GOSUB 1Ø81Ø
1ØØ6Ø IF CHOICE=2 THEN GOSUB 1Ø76Ø
1ØØ65 IF CHOICE=3 THEN GOSUB 1Ø3ØØ
1ØØ7Ø IF CHOICE=4 THEN RETURN
1ØØ75 IF CHOICE=5 THEN CLS:GOTO 2ØØ
1ØØ8Ø GOTO 1ØØ5Ø
1ØØ85 RETURN
1ØØ9Ø REM ---------- Load stock data to array ----------
1ØØ91 COUNT=Ø
1ØØ93 OPEN "i",2,SNAME$+".lst"
1ØØ95 WHILE NOT EOF(2)
1ØØ96    COUNT=COUNT+1
1ØØ97    INPUT#2,JD:INPUT#2,PPS:INPUT#2,SHARES
1Ø1ØØ    STOCK(2,COUNT)=PPS:STOCK(3,COUNT)=SHARES
1Ø1Ø4    STOCK(1,COUNT)=JD
1Ø11Ø WEND
1Ø115 CLOSE 2
1Ø117 IF COUNT>52 THEN BT=1:CT=COUNT-51 ELSE BT=53-COUNT:CT=1
1Ø12Ø RETURN
1Ø125 REM ---------- Update stock list ----------
1Ø14Ø GOSUB 132ØØ
1Ø145 OPEN "a",#2,SNAME$+".lst"
1Ø15Ø    PRINT#2,USING "#####";JD:PRINT#2,USING "###.##";PPS:PRINT#2,USING "####
.###";SHARES
1Ø2Ø5 CLOSE 2
1Ø21Ø RETURN
1Ø215 REM ---------- Load in current stock data ----------
1Ø22Ø OPEN "i",2,SNAME$+".stk"
1Ø225    INPUT#2,STOCKCOST:INPUT#2,SHARES
1Ø23Ø CLOSE 2
1Ø235 RETURN
1Ø3ØØ REM ---------- Print total stock history ----------
1Ø3Ø3 ESC$=CHR$(27):SUMPROD=Ø:SUMCOUNT=Ø:SUMSQR=Ø:COUNT=Ø:SUMPPS=Ø
1Ø3Ø5 OPEN "i",2,SNAME$+".lst"
1Ø31Ø    INPUT#2,NUM:INPUT#2,MIN:INPUT#2,NUM:MAX=MIN
1Ø315    WHILE NOT EOF(2)
1Ø32Ø       INPUT#2,NUM:INPUT#2,PPS:INPUT#2,NUM
1Ø325       IF PPS>=MAX THEN MAX=PPS ELSE IF PPS<MIN THEN MIN=PPS
1Ø326       COUNT=COUNT+1:SUMPROD=SUMPROD+COUNT*PPS:SUMCOUNT=SUMCOUNT+COUNT
1Ø327       SUMSQR=SUMSQR+COUNT^2:SUMPPS=SUMPPS+PPS
1Ø33Ø    WEND
1Ø335 CLOSE 2:AVE=(MAX+MIN)/2:IC=AVE*.Ø2:PC=.Ø2
1Ø337 LPRINT CHR$(14);TAB(2Ø-LEN(SNAME$));SNAME$:'print title double width
1Ø34Ø IF IC*3Ø=<(MAX-MIN) THEN IC=(MAX-MIN)/3Ø:PC=IC/AVE
1Ø345 LPPPS(15)=AVE
1Ø35Ø FOR A=15 TO 29:LPPPS(A+1)=INT((LPPPS(A)+IC)*1ØØ)/1ØØ:NEXT A
1Ø355 FOR A=15 TO 2 STEP -1:LPPPS(A-1)=INT((LPPPS(A)-IC)*1ØØ)/1ØØ:NEXT A
1Ø356 LPRINT:LPRINT:LPRINT TAB(3Ø);"Price Per Share ( ";INT(PC*1ØØ);"% )"
1Ø357 LPRINT ESC$;"A";CHR$(8);ESC$;"*";CHR$(Ø);:'set line feed 8/72"  copy font
to download ram
1Ø36Ø FOR A=1 TO 1Ø:'load new numbers
1Ø37Ø    FOR B=1 TO 14:LPRINT CHR$(ROTNUM(A,B));:NEXT B
1Ø375 NEXT A:LPRINT ESC$;CHR$(36);CHR$(1);:'select download character set
1Ø38Ø FOR A=1 TO 6
1Ø385    FOR B=1 TO 3Ø:LPRINT TAB(16+B*2);MID$(STR$(LPPPS(B)),A,1);:NEXT B
1Ø387    LPRINT
1Ø39Ø NEXT A:LPRINT ESC$;CHR$(36);CHR$(Ø);ESC$;"A";CHR$(6);:' cancel download ch
aracter set, set line feed 6/72"
```

```
1Ø395 LPRINT TAB(18);:FOR A=2Ø TO 8Ø:LPRINT CHR$(239);:NEXT A:LPRINT
1Ø4ØØ B=46-(2/IC)*AVE:SLOPE=(SUMPROD-SUMCOUNT*SUMPPS/COUNT)/(SUMSQR-SUMCOUNT^2/C
OUNT):INTERCEPT=AVE-SLOPE*SUMCOUNT/COUNT
1Ø4Ø2 COUNT=1
1Ø4Ø5 OPEN "i",2,SNAME$+".lst"
1Ø41Ø    WHILE NOT EOF(2)
1Ø415       INPUT#2,JD:INPUT#2,PPS:INPUT#2,SHARES:GOSUB 133ØØ
1Ø42Ø       LPRINT TAB(16);CHR$(239);:FOR A=18 TO 76 STEP 2:LPRINT TAB(A);".";:N
EXT A:LPRINT
1Ø425       LPRINT USING "& & & & &&";TAB(5);DAY$;MON$;YEAR$;:LPRINT TAB(16);CHR
$(239);
1Ø43Ø       FOR A=17 TO ((2/IC)*PPS+B):LPRINT TAB(A);CHR$(231);:NEXT A
1Ø432       YPOS=SLOPE*COUNT+INTERCEPT
1Ø435       LPRINT TAB((2/IC)*YPOS+B);CHR$(234)
1Ø44Ø       COUNT=COUNT+1
1Ø445    WEND:LPRINT TAB(16);CHR$(239)
1Ø45Ø CLOSE 2
1Ø455 LPRINT TAB(16);:FOR A=2Ø TO 8Ø:LPRINT CHR$(239);:NEXT A:LPRINT
1Ø47Ø LPRINT ESC$;"@":LPRINT:LPRINT:LPRINT:LPRINT:'reset printer
1Ø495 RETURN
1Ø76Ø REM ---------- Display purchases ----------
1Ø762 COUNT=CT
1Ø765 FOR A=BT TO 51
1Ø77Ø     FIRST=STOCK(3,COUNT)
1Ø775     SECOND=STOCK(3,COUNT+1)
1Ø78Ø     IF ABS(FIRST-SECOND)<5 THEN 1Ø797
1Ø785     XPOS=78+A*8
1Ø79Ø     YPOS=16
1Ø795     LINE (XPOS,176)-(XPOS,YPOS),4
1Ø797     COUNT=COUNT+1
1Ø8ØØ NEXT A
1Ø8Ø5 RETURN
1Ø81Ø REM ---------- Plot number of shares ----------
1Ø815 ' Plot right vertical scale
1Ø817 IF STOCK(3,COUNT-1)>1ØØØ THEN SCALE=5ØØ:EX=1 ELSE SCALE=5Ø:EX=Ø
1Ø82Ø FOR A=1 TO 2Ø
1Ø825     SYMBOL(492,173-A*8),"-",1,1,1,Ø
1Ø83Ø     SYMBOL(492,173-A*8),STR$(A*SCALE),1,1,6,Ø
1Ø835 NEXT A
1Ø84Ø LOCATE 7,68+EX:COLOR 6:PRINT "#"
1Ø845 LOCATE 9,68+EX:PRINT "S"
1Ø85Ø LOCATE 1Ø,68+EX:PRINT "H"
1Ø855 LOCATE 11,68+EX:PRINT "A"
1Ø86Ø LOCATE 12,68+EX:PRINT "R"
1Ø865 LOCATE 13,68+EX:PRINT "E"
1Ø87Ø LOCATE 14,68+EX:PRINT "S"
1Ø875 LOCATE 25,69:PRINT CHR$(196);CHR$(196);:COLOR 7:PRINT " # Shares";
1Ø88Ø ' Plot number
1Ø885 FLAG=Ø:COUNT=CT
1Ø89Ø FOR A=BT TO 52
1Ø9ØØ    XPOS=7Ø+A*8
1Ø9Ø5    YPOS=172-8*STOCK(3,COUNT)/SCALE
1Ø9Ø7    IF YPOS<12 THEN YPOS=12
1Ø91Ø    IF FLAG=1 THEN LINE(XPOS,LASTYPOS+4)-(XPOS,YPOS+4),6
1Ø915    IF A<52 THEN SYMBOL(XPOS,YPOS),CHR$(196),1,1,6,Ø
1Ø92Ø    LASTYPOS=YPOS:FLAG=1
1Ø922    COUNT=COUNT+1
1Ø925 NEXT A
1Ø93Ø RETURN
1Ø935 REM ---------- Draw weekly stock graph ----------
1Ø94Ø CLS:WINDOW(Ø,Ø)-(639,199)
1Ø945 VIEW(Ø,Ø)-(639,199),Ø,Ø
1Ø95Ø ' Draw heading
1Ø955 SYMBOL(136,1),"STOCK PRICE TREND",2,1,7,Ø
1Ø96Ø LOCATE 1,6Ø:COLOR 7:PRINT SNAME$
1Ø965 ' Draw graph outlines
1Ø97Ø LINE (72,176)-(72,16),2:LINE(71,176)-(71,16),2:LINE(7Ø,176)-(7Ø,16),2
1Ø975 LINE (488,176)-(488,16),2:LINE(489,176)-(489,16),2:LINE(49Ø,176)-(49Ø,16),
2
1Ø98Ø LINE (72,176)-(488,176),2:LINE(72,175)-(488,175),2
1Ø985 ' Draw left vertical scale
1Ø99Ø AVE=(MAX+MIN)/2
1Ø995 INCREMENT=AVE/1ØØ 'increment is 1% of ave
11ØØØ IF (MAX-MIN)/INCREMENT >8Ø THEN INCREMENT=INCREMENT+.Ø5:GOTO 11ØØØ:REM mak
e sure we do not exceed chart length
11ØØ5 COUNTINCREMENT=INCREMENT*4
11Ø1Ø IF AVE<1Ø THEN XPOS=24 ELSE XPOS=16
11Ø15 SYMBOL (XPOS,93),STR$(INT(AVE*1ØØ)/1ØØ),1,1,3,Ø
```

```
11020 SYMBOL(64,93),"-",1,1,1,0
11025 LINE(74,96)-(486,96),5
11030 VALUE=INT(AVE*100)/100:YPOS=93
11035 WHILE YPOS<=160
11040     VALUE=VALUE-COUNTINCREMENT
11045     YPOS=YPOS+8
11050     IF VALUE<10 THEN XPOS=24 ELSE XPOS=16
11055     SYMBOL (XPOS,YPOS),STR$(INT(VALUE*100)/100),1,1,3,0
11060     SYMBOL (64,YPOS),"-",1,1,1,0
11065     LINE (74,YPOS+3)-(486,YPOS+3),5
11070 WEND
11075 MINVALUE=VALUE
11080 VALUE=INT((AVE+.005)*100)/100:YPOS=93
11085 WHILE YPOS>16
11090     VALUE=VALUE+COUNTINCREMENT
11095     YPOS=YPOS-8
11100     IF VALUE<10 THEN XPOS=24 ELSE XPOS=16
11105     SYMBOL (XPOS,YPOS),STR$(INT(VALUE*100)/100),1,1,3,0
11110     SYMBOL (64,YPOS),"-",1,1,1,0
11115     LINE (74,YPOS+3)-(486,YPOS+3),5
11120 WEND
11125 ' Draw breakevan line
11130 YPOS=168-2*(STOCKCOST/SHARES-MINVALUE)/INCREMENT
11135 LINE(74,YPOS)-(486,YPOS),1
11140 ' Draw left vertical label
11145 LOCATE 4,1:COLOR 3:PRINT"P"
11150 LOCATE 5,1:PRINT"R"
11155 LOCATE 6,1:PRINT"I"
11160 LOCATE 7,1:PRINT"C"
11165 LOCATE 8,1:PRINT"E"
11170 LOCATE 10,1:PRINT"P"
11175 LOCATE 11,1:PRINT"E"
11180 LOCATE 12,1:PRINT"R"
11185 LOCATE 14,1:PRINT"S"
11190 LOCATE 15,1:PRINT"H"
11195 LOCATE 16,1:PRINT"A"
11200 LOCATE 17,1:PRINT"R"
11205 LOCATE 18,1:PRINT"E"
11210 LOCATE 23,69:PRINT CHR$(196);CHR$(196);:COLOR 7:PRINT " PPS"
11215 LOCATE 24,69:COLOR 1:PRINT CHR$(196);CHR$(196);:COLOR 7:PRINT " BE"
11220 ' Draw control box
11225 LINE (556,40)-(638,168),2,B
11230 LOCATE 5,71:COLOR 7:PRINT "OPTIONS:"
11235 LOCATE 7,71:PRINT "1-Display"
11240 LOCATE 8,71:PRINT "  Shares"
11245 LOCATE 10,71:PRINT "2-Display"
11250 LOCATE 11,71:PRINT "Purchases"
11255 LOCATE 13,71:PRINT "3-Print"
11260 LOCATE 14,71:PRINT "  Data"
11265 LOCATE 16,71:PRINT "4-Next"
11270 LOCATE 17,71:PRINT "  Chart"
11275 LOCATE 19,71:COLOR 4:PRINT "5-Menu"
11280 RETURN
11285 REM ---------- Plot pps chart ----------
11290 FLAG=0:COUNT=CT
11295 FOR A=BT TO 52
11305    YPOS=164-2*(STOCK(2,COUNT)-MINVALUE)/INCREMENT
11310    XPOS=70+A*8
11315    IF FLAG=1 THEN LINE(LASTXPOS,LASTYPOS+4)-(XPOS,YPOS+4),3
11322    JD=STOCK(1,COUNT):GOSUB 13300
11324    IF LEN(DAY$)=2 THEN D$="0" ELSE D$=MID$(DAY$,2,1)
11325    SYMBOL(XPOS-2,177),D$,1,1,7,0
11330    SYMBOL(XPOS-2,185),RIGHT$(DAY$,1),1,1,7,0
11335    SYMBOL(XPOS-2,193),LEFT$(MON$,1),1,1,6,0
11340    LASTYPOS=YPOS:FLAG=1:LASTXPOS=XPOS
11342    COUNT=COUNT+1
11345 NEXT A
11350 RETURN
11355 REM ---------- Draw right vertical scale ----------
11360 FOR A=1 TO 20
11365     SYMBOL(488,176-A*8),"-",1,1,5,0
11370     SYMBOL(496,176-A*8),50*A-50,1,1,5,0
11375 NEXT A
11380 RETURN
11435 REM ---------- Input Price-Per-Share ----------
11440 FOR A=0 TO STKNUM
11445    LOCATE 10+A,5:COLOR 2:PRINT USING "## &       &";A;NAMES$(A);
11450    COLOR 7:INPUT PPS(A)
11455 NEXT A
```

```
11460 LOCATE 24,1:COLOR 4:INPUT "Does input data look correct,y/n? ",A$
11465 WHILE A$="n"
11470    LOCATE 24,1:PRINT SPACE$(50)
11475    LOCATE 24,1:COLOR 2:INPUT "Enter the number of the bad price-per-share.
 ",BADNUM
11480    LOCATE 10+BADNUM,15:PRINT SPACE$(50)
11485    LOCATE 24,1:PRINT SPACE$(50);TAB(1);"Enter price-per-share. "
11490    LOCATE 10+BADNUM,15:INPUT PPS(BADNUM)
11495    LOCATE 24,1:PRINT SPACE$(50);TAB(1);"Does input data look correct,y/n?
";
11497    COLOR 7:INPUT A$
11500 WEND
11505 RETURN
12000 REM ---------- Draw empty bar graph ----------
12005 CLS:WINDOW (0,0)-(639,199)
12010 VIEW (1,1)-(639,199),0,0
12015 GOSUB 12350
12020 REM ******DRAW CHART BASE
12025 LINE(60,180)-(576,180),1
12030 LINE(76,176)-(576,176),1
12035 LINE(576,180)-(576,176),1
12040 LINE(60,180)-(76,176),1
12045 LINE(576,180)-(592,176),1
12050 PAINT(70,179),6,1
12055 REM ******DRAW LEFT CHART SIDE
12060 LINE(60,180)-(60,20),1
12065 LINE(76,176)-(76,16),1
12070 LINE(60,20)-(76,16),1
12075 PAINT(61,179),6,1
12080 REM ******DRAW RIGHT CHART SIDE
12085 LINE(576,180)-(576,20),1
12090 LINE(592,176)-(592,16),1
12095 LINE(576,20)-(592,16),1
12100 PAINT(578,178),6,1
12105 GOSUB 12370:REM ******draw vertical hatching
12110 REM ******DRAW VERTICAL SCALE
12115 SYMBOL(45,177),"0-",1,1,1,0
12120 FOR A%=1000 TO 8000 STEP 1000:SYMBOL (13,177-(A%/50)),STR$(A%)+"-",1,1,1,0
:NEXT A%
12125 SYMBOL(1,50),"D",1,1,7,0
12130 SYMBOL(1,60),"O",1,1,7,0
12135 SYMBOL(1,70),"L",1,1,7,0
12140 SYMBOL(1,80),"L",1,1,7,0
12145 SYMBOL(1,90),"A",1,1,7,0
12150 SYMBOL(1,100),"R",1,1,7,0
12155 SYMBOL(1,110),"S",1,1,7,0
12160 SYMBOL(608,20),"Ü",1,1,3,0:SYMBOL(613,28),"-COST",1,1,7,1
12165 SYMBOL(608,80),"Ü",1,1,2,0:SYMBOL(613,88),"-VALUE",1,1,7,1
12170 RETURN
12175 REM ---------- Display data on bar graph ----------
12180 IF VALUE<=0 THEN RETURN
12185 IF FIRSTPASS>0 THEN A$="2":FIRSTPASS=0
12190 ON VAL(A$) GOTO 12195,12210,12230
12195    IF VALUE<20 THEN VALUE=20
12200    IF VALUE>800 THEN VALUE=800
12205 GOTO 12245
12210    IF VALUE<200 THEN VALUE=200
12215    IF VALUE>8000 THEN VALUE=8000
12220    VALUE=VALUE/10
12225 GOTO 12245
12230    IF VALUE<2000 THEN VALUE=2000
12235    IF VALUE>80000! THEN VALUE=80000!
12240    VALUE=VALUE/100
12245 VALUE=180-VALUE/5
12250 REM ******DRAW BAR OUTLINE
12255 LINE(BAR+12,180)-(BAR+24,176),CL
12260 LINE(BAR,VALUE)-(BAR+12,VALUE-4),CL
12265 LINE(BAR+12,VALUE-4)-(BAR+24,VALUE-4),CL
12270 LINE(BAR,180)-(BAR+12,180),CL
12275 REM ******DRAW LEFT AND RIGHT VERTICALS
12280 LINE(BAR,180)-(BAR,VALUE),CL
12285 LINE(BAR+24,176)-(BAR+24,VALUE-4),CL
12290 REM ******PAINT IN BAR COLOR
12295 PAINT (BAR+8,179),CB,CL:PAINT (BAR+22,175),CB,CL
12300 LINE(BAR+1,176)-(BAR+22,176),CB
12305 REM ******DRAW BAR OUTLINES
12310 LINE(BAR,VALUE)-(BAR+12,VALUE),CL
12315 LINE(BAR+12,VALUE)-(BAR+24,VALUE-4),CL
12320 LINE(BAR+12,VALUE)-(BAR+12,180),CL
```

```
12325 LINE(BAR,18Ø)-(BAR+14,18Ø),1
1233Ø RETURN
12335 REM ---------- Draw labels on bar graph ----------
1234Ø SYMBOL(BAR-12,183),LABEL$,1,1,7,Ø
12345 RETURN
1235Ø REM ---------- Draw bar graph heading ----------
12355 SYMBOL(1ØØ-LEN(HEADING$)/2,1),"ÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜ",2,1,Ø,Ø
1236Ø SYMBOL(1ØØ-LEN(HEADING$)/2,1),HEADING$,2,1,7,Ø
12365 RETURN
1237Ø REM ---------- Draw vertical hatching ----------
12375 FOR A=17Ø TO 2Ø STEP -1Ø:LINE(6Ø,A)-(76,A-4),1:LINE(78,A-4)-(575,A-4),7:NE
XT A
1238Ø LINE(576,16)-(591,16),7
12385 RETURN
1239Ø REM ---------- Load in stock names ----------
12395 OPEN "i",2,"stknames.dat"
124ØØ    INPUT#2,STKNUM
124Ø5    FOR A=Ø TO STKNUM
1241Ø       INPUT#2,NAMES$(A)
12415    NEXT A
1242Ø CLOSE 2
12425 RETURN
1243Ø REM ---------- Draw scale Ø-8ØØ ------------
12435 LINE(78,16)-(576,175),Ø,BF:GOSUB 1237Ø
1244Ø FOR A=1ØØØØ TO 8ØØØØ! STEP 1ØØØØ:SYMBOL (8,177-(A/5ØØ)),"ÜÜÜÜÜÜÜ",1,1,Ø,Ø:N
EXT A
12445 FOR A%=1ØØ TO 8ØØ STEP 1ØØ:SYMBOL (21,177-(A%/5)),STR$(A%)+"-",1,1,1,Ø:NEX
T A%
1245Ø RETURN 53Ø
12455 REM ---------- Draw scale Ø-8ØØØØ ----------
1246Ø LINE(78,16)-(576,175),Ø,BF:GOSUB 1237Ø
12465 FOR A=1ØØØØ TO 8ØØØØ! STEP 1ØØØØ:SYMBOL (8,177-(A/5ØØ)),"ÜÜÜÜÜÜÜ",1,1,Ø,Ø:N
EXT A
1247Ø FOR A=1ØØØØ TO 8ØØØØ! STEP 1ØØØØ:SYMBOL (8,177-(A/5ØØ)),STR$(A),1,1,1,Ø:NE
XT A
12475 RETURN 53Ø
1248Ø REM ---------- Draw scale Ø-8ØØØ ----------
12485 LINE(78,16)-(576,175),Ø,BF:GOSUB 1237Ø
1249Ø FOR A=1ØØØØ TO 8ØØØØ! STEP 1ØØØØ:SYMBOL (8,177-(A/5ØØ)),"ÜÜÜÜÜÜÜ",1,1,Ø,Ø:N
EXT A
12495 FOR A%=1ØØØ TO 8ØØØ STEP 1ØØØ:SYMBOL (13,177-(A%/5Ø)),STR$(A%)+"-",1,1,1,Ø
:NEXT A%
125ØØ RETURN 53Ø
125Ø5 REM ---------- Wait for keypressed ----------
1251Ø FOR A=1 TO 5ØØ
12515    A$=INKEY$
1252Ø    IF A$<>"" THEN RETURN 64Ø
12525 NEXT A
1253Ø RETURN
13ØØØ REM ---------- Update purchases/sale list ----------
13ØØ3 GOSUB 132ØØ:' calculate date
13ØØ4 ON ERROR GOTO 141ØØ
13ØØ5 OPEN "i",2,NAMES$(VAL(B$))+".lst":OPEN "o",3,"TEMPFIL.LST":F=Ø:' sort by d
ate
13ØØ8 WHILE NOT EOF(2)
13Ø1Ø    INPUT#2,SJD:INPUT#2,SPPS:INPUT#2,SSHARES
13Ø11    IF SJD=JD AND F=Ø THEN SPPS=PPS:F=1
13Ø14    IF SJD>JD AND F=Ø THEN PRINT#3,USING "#####";JD:PRINT#3,USING "###.##";
PPS:PRINT#3,USING "####.###";NEWSHARES:F=1
13Ø15    IF F=1 THEN SSHARES=NEWSHARES
13Ø16    PRINT#3,SJD:PRINT#3,SPPS:PRINT#3,SSHARES
13Ø18 WEND:CLOSE 2
13Ø19 IF F=Ø THEN PRINT#3,USING "#####";JD:PRINT#3,USING "###.##";PPS:PRINT#3,US
ING "####.###";NEWSHARES
13Ø2Ø CLOSE 3
13Ø21 OPEN "i",2,"tempfil.lst":OPEN "o",3,NAMES$(VAL(B$))+".lst"
13Ø22 WHILE NOT EOF(2)
13Ø23    INPUT#2,SJD:INPUT#2,SPPS:INPUT#2,SSHARES
13Ø24    PRINT#3,USING "#####";SJD:PRINT#3,USING "###.##";SPPS:PRINT#3,USING "##
##.###";SSHARES
13Ø26 WEND:CLOSE 2:CLOSE 3
13Ø45 OPEN "a",2,NAMES$(VAL(B$))+".pur"
13Ø55    PRINT#2,L$:PRINT#2,USING "#####";JD:PRINT#2,USING "####.##";PPS:PRINT#2
,USING "####.###";SHARESC:PRINT#2,USING "######.##";STOCKC
13Ø65 CLOSE 2
13Ø7Ø RETURN
13Ø75 REM ---------- Dividend purchase ----------
13Ø8Ø LOCATE 6,1:PRINT SPACE$(55):LOCATE 6,1:INPUT "Enter price-per-share of sto
ck purchased. ",PPS
```

A-OK Computers

The A-OK Board™—The Ultimate Upgrade™
"10 MHz + 960K"

The A-OK Board lets you triple (!) your processor speed and nearly quadruple your RAM. **The A-OK Board** plugs inside your Sanyo with no soldering or trace cutting (either would void your Sanyo motherboard's 1 year warranty). **The A-OK Board**'s 8088-1 does all your processing at 10 MHz, almost three times the speed. Three banks of 256K chips on **The A-OK Board** combined with 192K on your motherboard gives you 960K of contiguous RAM. **The A-OK Board** has a socket to accept a highly compatible ROM BIOS. See our **TurboDrive550** ad about 800K ramdisks. Options include a real-time clock/calendar. The crystal and 8088-1 are socketed for easy upgrade to faster processors as they become available. **The A-OK Board** is unique in that it increases processing speed without driving the motherboard faster. Running the motherboard faster can severely reduce the lifetime of chips soldered in the motherboard. **The A-OK Board** *is* **The Ultimate Upgrade.**

A-OK Board with no memory $185
8088-1 10 MHz processor chip . . . . . $ 25
Clock/Calendar for **The A-OK Board** . . . . . $ 69
A-OK Board with "The Works" $395

TP*Linker
"The Linker for Turbo Pascal"

TP*Linker accelerates the compilation of your Turbo Pascal 2.0 and 3.0 programs. **TP*Linker** provides a program that turns your debugged and compiled Turbo Pascal procedures into compact, linkable external procedures. Because they are already compiled, these procedures link instantly into the code you are debugging. **TP*Linker**'s linkable procedures use much less memory than the Turbo Pascal source code they come from. Thus you can bring them in off disk faster and store more of them on RAM disk. **TP*Linker** is easy to use, self-documenting, and works from the Turbo Pascal Program Development Environment. An extra bonus—now you can sell procedures written in Turbo Pascal as linkable procedures without having to disclose your source code. A-OK is looking for high-quality Turbo Pascal libraries to market. Why waste time recompiling code that you know works? **TP*Linker** is the ideal upgrade for Turbo Pascal.

TP*Linker for Turbo Pascal 2.0 and 3.0 . . . . . $69
TP*Linker for Turbo Pascal 2.0, 3.0, BCD, and 8087 . . . . . $99

SANYGRAF
"Professional Business Graphics"

A-OK searched the world to bring you the best business graphics for your Sanyo. We are delighted by what we found—**SANYGRAF. SANYGRAF**'s well thought-out menus and superb documentation, make this extremely versatile software very easy to master. **SANYGRAF** is *very* fast and even faster with an 8087. **SANYGRAF** supports popular plotters. **SANYGRAF** is so good that Sanyo Australia supplies it with every color graphics system it sells. 2.11, 256K, one DSDD or DSQD drive required.

SANYGRAF $99

Janus/Ada
"The Ada for Micros"

Now you can use the language of the future on your Sanyo, IBM, Zenith, etc. For just $99, you get the **Janus/Ada** Compiler, Linker, Libraries, Example Programs, User Manual, and User Group Membership! Great for learning Ada and for serious programming.

A-OK is pleased to announce its selection as the only distributor of **Janus/Ada** on the exclusive Federal Government's GSA Schedule. Our contract number is GS00K85AGS6083.

Janus/Ada C-Pak $99

Janus/Ada Extended Tutorial

This extraordinary package makes learning Janus/Ada easy and fun. Each of the handbook's 10 chapters guides you to an understanding of a different aspect of **Janus/Ada.** The 150+ page handbook is complemented by examples and quizzes on disk. This is an excellent tool for learning **Janus/Ada.**

Janus/Ada Extended Tutorial $99

TurboDrive550™
"A Quality Ram Disk for Sanyo"

TurboDrive550 turns your unused RAM into a disk drive.
TurboDrive550 is very fast.

- Speeds up all disk intensive software.
- Makes WordStar and FORMSORT run more quickly.
- Accesses disk 75,000 times faster than hard disk.
- Makes 800K ramdisks on the **A-OK Board.**
- Using the new **A-OK DOS-3** with your single or double-sided drives.
- Using the new **A-OK DOS-4** with your quad drives.

TurboDrive550 is a low-cost alternative to a second disk drive.
Compare **TurboDrive550's** features to the competition.

- **TurboDrive550** works even if you only have 128K.
- **TurboDrive550** protects you from allocating more space than is available.
- A-OK provides a COM file in MS-DOS 1.25 which lets you gracefully free **TurboDrive550**'s space.
- **TurboDrive550** lets you use its menu or parameters.

TurboDrive 550 is The Phantom Buster.

TurboDrive550 comes **FREE** with every new **A-OK DOS 1-4™** and is available as a $7 upgrade to current **A-OK DOS** owners.
TurboDrive550 for other Sanyo DOSs $33
(The best deal is to buy **A-OK DOS-1™** for $35 and get **TurboDrive550** free.)

Contact!
"Mailing Lists and More"

Contact! makes it incredibly easy for you to create and maintain business and personal mailing lists. **Contact!** prints mailing labels in the order you want and creates files ready for MailMerge. **Contact!** even prints out your address book complete with phone numbers. Requires 2.11 or Video Board.

Contact! $45

N-Code™
"If You are Serious about Data Security"

N-Code for the Sanyo MBC-550 Series or IBM PC $79

"AMBIZ-PAK"™

Fifteen programs for Amway Product Distributors **$100**

UPGRADE TO 800K PER DRIVE

Increase your disk capacity to 800K per drive using A-OK's **DSQD Kit™** which includes: 2 first quality new TEAC 55F double-sided quad-density (DSQD) drives which replace the TEACs in your Sanyo, **A-OK DOS-4**, complete easy instructions, and all parts needed.

DSQD Kit $355

DSDD Kit™

Upgrade to 400K per drive—same as DSQD Kit but with 2 TEAC 54B drives and **A-OK DOS-2.**

DSDD Kit $295

VidSwitch550™

Just push a button on the left front side of your Sanyo and you've switched from the Sanyo graphics port to the Video Board port. Another push of the button on this small box switches it back. The monochrome version requires only the mono cable you already have. The color version requires no extra cables. Everything just plugs in.

VidSwitch550—Monochrome $29
VidSwitch550—Color (with sync inversion) $69

XCord550™
Extension cord for your Sanyo PC Keyboard

6 foot **$15**

RS232C Serial Boards $72

THE A-OK TEAM

Join the team that created the **A-OK DOS** Family—the choice of OEMs, **PrintScreen550, TurboDrive550, The A-OK Board** and other top-quality software and hardware. We sell only the best. Fewer titles, better promotion. Give us a call. Join the **A-OK Team.**

SEE OUR OTHER FULL-PAGE AD FOR ORDERING INSTRUCTIONS.

DEALER, DISTRIBUTOR, AND OEM INQUIRIES INVITED

A-OK Computers 816 Easley St., Suite 615 • Silver Spring, MD 20910 301-585-5105, 301-585-5106

```
13085 NEWSHARES=VAL(MID$(NEWSHARES$,2,LEN(NEWSHARES$)))
13087 L$="DP"
13090 RETURN
13095 REM ---------- Stock purchase ------------
13100 NEWSHARES=VAL(NEWSHARES$)
13105 LOCATE 7,1:COLOR 2:PRINT"Enter amount paid for stock. "
13110 LOCATE 7,30:COLOR 7:INPUT STOCKAMOUNT
13115 PPS=STOCKAMOUNT/NEWSHARES
13117 L$="P"
13120 RETURN
13125 REM ---------- Stock sale -----------
13130 LOCATE 5,1:PRINT SPACE$(55):LOCATE 5,1:INPUT "Enter number of shares, pric
e per share.  ",NEWSHARES,PPS
13135 STOCKAMOUNT=-1*NEWSHARES*PPS:NEWSHARES=NEWSHARES*-1
13137 L$="S"
13140 RETURN 770
13200 REM ---------- Encode julian date ----------
13205 POS1=INSTR(DATE$,"-")
13210 MON=VAL(LEFT$(DATE$,POS1-1))
13215 POS2=INSTR(POS1+1,DATE$,"-")
13220 DAY=VAL(RIGHT$(DATE$,POS2+1))
13222 YEAR=VAL(RIGHT$(DATE$,2))
13225 JD=MONLEN(MON)+DAY+YEAR*1000
13230 RETURN
13300 REM ---------- Decode julian date ----------
13305 JD$=STR$(JD)
13310 JD=VAL(MID$(JD$,4,LEN(JD$)))
13315 YEAR$=MID$(JD$,2,2)
13320 FOR D=12 TO 1 STEP -1
13325    IF JD>MONLEN(D) THEN MON$=MON$(D):DAY$=STR$(JD-MONLEN(D)):RETURN
13330 NEXT D
13335 RETURN
13400 REM ---------- Load arrays ----------
13402 RESTORE
13405 FOR A=1 TO 10
13410    FOR B=4 TO 14
13412       ROTNUM(A,1)=27:ROTNUM(A,2)=42:ROTNUM(A,3)=1
13415       READ ROTNUM(A,B)
```

A-OK Computers

The Center of Creation and Dissemination of Sanyo PC Software and Hardware

The A-OK™ DOS Family

"Upgrade your MS-DOS 1.25 or 2.11 and Video Board"

All members of the **A-OK DOS** family add these features to your MS-DOS 1.25 or 2.11:

- **A-OK DOS** supports RAM expansion to 960K.

With the Video Board, **A-OK DOS** uses 24K *less* memory than the "competition". With extended memory, **A-OK DOS** uses 55K *less* memory than the "competition".

- **A-OK DOS** now includes *free* high-quality ramdisk software—**TurboDrive550** (a $33 value).
- Still the only text dump for 2.11 automatically transferred via FORMAT /S.
- Prints a screenful of text to any Sanyo-compatible printer.
  - Interactive mode
    - It works with one keystroke in DOS, BASIC and all other software packages that run on the Sanyo.
    - Does not interfere with program execution or affect the data.
  - Batch mode
    - Can be activated in BASIC by a one-line command.
    - Interrupt-driven and can be called from programming languages.
- Supports the only graphics screen dump software built into DOS—**PrintScreen550.**
- Includes a special "Sorted Directory" utility.
- Has a utility to strip the high-order bit off WordStar document files.
- User-controlled scrolling in all subsystems, not just DOS.
- **A-OK DOS** system messages are easier to understand.
- Supports up to 4 disk drives.

A-OK DOS versions for MS-DOS 1.25 correct flaws in MS-DOS 1.25 without sacrificing its strengths.

- All **A-OK DOS/1.25** are much faster and use far less memory than any version of MS-DOS 2.11.
- Runs all software that comes with the Sanyo.

A-OK DOS versions for MS-DOS 2.11:

- Run all software which runs on the Sanyo under MS-DOS 2.11 and Video Board.
- PLUS special support for Sidekick™ and Flight Simulator.™
- Support installable drivers for hard disks, etc.

Comparing **A-OK DOS-3** and DS DOS+, the *Sanyo PC Hackers Newsletter* states "In a nutshell, you get more for your money in A-OK's package". A-OK has tailored different versions of its DOS for varying disk drive configurations. This makes our ad more complex, but our DOS more efficient and easier for you to use.

A-OK DOS-1™

"Not just another Double-Sided DOS"

- Any combination of up to four single and double-sided drives.
- Double-sided drives support single and double-sided diskettes.
- 8 and 9 sector formats—360K, 320K, 180K, and 160K.

A-OK DOS-1/1.25 or **A-OK DOS-1**/2.11 and Video Board **$35**

A-OK DOS-2™

"Increase the Capacity of your Current Drives"

- All the features and formats of **A-OK DOS-1**
- PLUS it reads, writes, and formats 10-sector formats
  - 400K per double-sided diskette on DSDD drives
  - 200K per single-sided diskette on SSDD and DSDD drives
  - uses ordinary diskettes
  - pays for itself in saved diskettes.

A-OK DOS-2/1.25 or **A-OK DOS-2**/2.11 and Video Board **$45**

800K PER DISK DRIVE

A-OK DOS-3™

"**A-OK DOS** for Double and Quad-Density Drive Systems"

- All the features of **A-OK DOS-2**
- PLUS up to 4 double-density and quad-density drives
  - on double-density drives:
    - reads and writes all formats read by 1.25 and 2.11
    - reads, writes, and formats 200K SSDD diskettes
    - reads, writes and formats 400K DSDD diskettes
  - on quad-density drives:
    - reads, writes, and formats 730K and 800K on ordinary diskettes.

A-OK DOS-3/1.25 or **A-OK DOS-3**/2.11 and Video Board **$55**

A-OK DOS-4™

"The *Standard* Quad-Density DOS"

- All the features of **A-OK DOS-2**
- PLUS up to 4 quad drives—3.2 Megabytes of *removable* medium!
  - quad drives read all diskettes written by MS-DOS 1.25 and 2.11
  - quad drives read, write and format 160K, 180K, 200K, 320K, 360K, 400K, 730K, 800K formats on ordinary diskettes.
  - will not run Prolok™ disks.

We at A-OK Computers thank you for making **A-OK DOS-4**™ *the* standard DOS for quad-density drives on the Sanyo. Thousands of copies of **DOS-4**™ are now in use in the most demanding applications—yours. Potential bugs have been identified and corrected, making **DOS-4** a mature, well seasoned tool. Your constant feedback has resulted in the successive refinement of our documentation. After thorough testing and comparison to would-be competitors, Scottsdale Systems, America's No. 1 seller of Sanyos, selected **DOS-4** as the standard DOS supplied with their Silver Fox™. (HAGEN-DOS™ is A-OK DOS-4). And of course, MidWest Micro Peripherals—one of America's largest sellers of computer equipment has selected **A-OK DOS-4** for their quad-based Sanyo. Because **DOS-4** is now the standard, our 720K and 800K formats are also regarded as the standard. Of the quad-density operating systems, only **A-OK DOS-4** supports the very popular **PrintScreen550.** *Soft Sector* sums it up well in their September 1984 review entitled "A-OK DOS and PrintScreen550—An Unbeatable Combination".

A-OK DOS-4 is the choice of OEMs. Don't trust your work to an experimental DOS, be sure your operating system is A-OK.

Look for these features in a quad-density drive DOS:

1. Unique support for Lotus 1-2-3™ and Picasso™ on quad drives.
2. Still the *only* DOS which you can install with quad drives.
3. The system *automatically* recognizes 8/9/10 sector, 40 and 80 track formats without reconfiguration or using alternate drive symbols (e.g. E:, F:, etc.).
4. Free ramdisk—**TurboDrive550**™.
5. Uses low-cost ordinary double-density diskettes.
6. FORMATs, READs, WRITEs, and DISKCOPYs *all* MS-DOS standard formats employed by the Sanyo 550 Series.
7. Uses the standard **DOS-4** 730K and 800K formats.
8. Built-in text screen to printer dump.
9. Supports the *only* graphics screen dump software built into the DOS-**PrintScreen550.**

A-OK DOS-4/1.25 or **A-OK DOS**/2.11 and Video Board $55

PrintScreen550™

"Fast, High-Resolution Graphics and Text from Screen to Printer with One Keystroke"

- **PrintScreen550** now also supports the Video Graphics Board.
- **PrintScreen550** is designed for the everyday needs of most users.
- **PrintScreen550** capabilities
  - Prints a screenful of graphics to most popular printers.
    - This prints everything on the screen whether it's graphics or text.
    - User has the option to rotate the image on the paper. This permits Okidata 92s and other printers with less than 640 horizontal dot positions to print out the full screen.
    - Turns your Sanyo into a high-resolution graphics machine.
      - Multiscreen facility permits a printed image to be 640 dots by an almost unlimited number of dots in the perpendicular direction.
- **PrintScreen550** is flexible and easy to use.
  - Interactive mode
    - Works like the "Print Screen" button on the IBM-PC.
    - One keystroke does it all. No need to embed commands in BASIC. No extra programs to execute. Same single keystroke works in DOS, BASIC and all other software packages that run on the Sanyo.
  - Batch mode can be activated in BASIC by a one-line command.
- **PrintScreen550** is very fast and a lot of fun!

A-OK's PrintScreen550 software pulls the Print Screen button off the IBM-PC and puts it on your Sanyo MBC-550 Series computer.

PrintScreen550 is an option to any A-OK DOS for a meager **$24**

Orders may be placed immediately.

Please send money order or cashiers check. Personal or company checks require 2-3 weeks to clear. Prices reflect a 3% cash discount. MasterCard/VISA/CHOICE accepted. MD residents add 5% sales tax. Out-of-state order, no tax. Prices subject to change. All brands are registered trademarks. 20% restocking fee for all returned merchandise. No returns accepted without a Return Authorization Number.

A-OK Computers 816 Easley St., Suite 615 • Silver Spring, MD 20910 301-585-5105, 301-585-5106

```
13420    NEXT B
13425 NEXT A
13430 FOR A=1 TO 12
13435    READ MONLEN(A),MON$(A)
13440 NEXT A
13445 DATA 49,0,62,0,8,0,8,0,14,0,8,50,0,126,0,4,8,16,32,0,66,60,51,0,126,64,32,
16,8,16,32,64,126
13450 DATA 52,0,32,0,32,94,32,4,32,8,48,53,0,62,64,0,64,62,0,2,0,126,54,0,60,66,
0,66,60,2,4,8,16
13455 DATA 55,0,4,0,8,0,16,0,32,0,126,56,0,60,66,0,66,60,66,0,66,60,57,0,8,16,32
,0,60,66,0,66,60
13460 DATA 48,0,24,36,66,0,66,0,66,36,24
13465 DATA 0,"Jan",31,"Feb",59,"Mar",90,"Apr",120,"May",151,"Jun",181,"Jul",212,
"Aug",243,"Sep",273,"Oct",304,"Nov",334,"Dec"
13470 RETURN
13500 REM ---------- Print purchase/sale list ----------
13505 ESC$=CHR$(27)
13510 LPRINT TAB(20);"Record of stock transactions"
13515 LPRINT TAB(20);"for ";ESC$;CHR$(69);NAMES$(VAL(B$))
13520 LPRINT ESC$;CHR$(70):LPRINT:LPRINT
13521 LPRINT "Action";TAB(15);"Julian Date";TAB(33);"PPS";TAB(49);"Shares";TAB(6
4);"Amount"
13525 OPEN "i",2,NAMES$(VAL(B$))+".pur"
13530    WHILE NOT EOF(2)
13535       INPUT#2,C$:INPUT#2,JD:INPUT#2,PPS:INPUT#2,SHARES:INPUT#2,STOCKAMOUNT
13540       LPRINT C$,JD,PPS,ABS(SHARES),ABS(STOCKAMOUNT)
13545    WEND
13550 CLOSE 2
13555 RETURN
14000 REM ---------- First run ----------
14005 STKNUM=-1
14010 RESUME 950
14100 REM ---------- File .lst not found ----------
14105 OPEN "o",2,NAMES$(VAL(B$))+".lst"
14110    PRINT#2, USING "#####";JD:PRINT#2, USING "###.##";PPS:PRINT#2, USING "#
###.###";NEWSHARES
14115 CLOSE 2
14120 RESUME 13045
```

Master Menu

By Dale E. Baker

"All right, got a new BASIC program. I'll just put it on this disk with about 100 other programs. Say, what's this program doing here, and what the heck does it do? So *this* is where I hid that utility! "One of these days I'll get organized yet. Think I'll take this menu program that's been floating 'round and try to put these programs in some order, but let's see, I'll have to change this, and if I do that, then this won't work!"

Sound familiar? I finally got tired of modifying menus to fit my programs and disks, so I put together a menu program that easily adapts to new programs and includes a brief description of each program, because I could no longer remember what *DEMO.BAS* does. *Master Menu* to the rescue! It allows

Dale Baker is a medic in the United States Air Force, currently stationed at Altus Air Force Base in Oklahoma. He purchased his first computer, a Sanyo 555, in Tokyo, Japan three years ago and has never regretted it. He programs in dBASE II and plays around with BASIC.

you to create nine separate menus on a disk with up to 26 programs controlled by each menu. The reason for those limits is that I wanted to press only a single key to run a program (necessity may be the mother of invention, but laziness is most assuredly the father). One could add 26 more programs to each menu by using lowercase letters, if desired.

Master Menu is written in blocks, as indicated by the remarks, each block performs a specific function then returns to the controlling block. I don't know if this is good programming, but it works. After typing in and debugging the program, you'll want to see how it runs, so I'll talk you through the blocks.

When the program is first run, it looks for a data file titled MENU.1; if this file doesn't exist it lets you know and takes you to the menu generation routine. For this reason, always create MENU.1 as your first menu — all you have to do is enter the number and *Master Menu* does the rest.

- Block 1, lines 10-60: Nothing complex here. Line 40 sets up function key 11 (CTRL-1) to return to *Master Menu* after running a program. I chose key 11 because a lot of programs I use make use of the 10 normal function keys.
- Block 2, lines 80-310: Line 120 takes us to Block 5 to check for existence of MENU.1; if it doesn't exist then it takes to blocks 4, 7 and 8. The rest of Block 2 draws the menu on the screen.
- Block 3, lines 310-440: This is the controlling portion, where you tell *Master Menu* you want to run a program, load a different menu, create/modify an existing menu or quit.
- Block 4, lines 460-600: This subroutine tells what menus are on the disk and lets you decide which menu to create or modify.
- Block 5, lines 620-760: Each menu, MENU.1 for example, is really just a data file for *Master Menu*. This subroutine opens the chosen menu and loads the menu's title, with up to 26 descriptions and 26 programs into variables.
- Block 6, lines 780-1040: Displays the information loaded in Block 5 and gives you the option of making changes or not.
- Block 7, lines 1060-1400; First, the menu's title is displayed and you are offered the opportunity to change it. Next, each description and the program it goes with is displayed. You may press CR (the RETURN key) to skip to the next description/program, '*' to make changes or ESC to return to the main menu.
- Block 8, lines 1420-1540: Block 7 runs into Block 8. This block deletes the old version of the menu and stores the new version on disk.

Other Notes

Because *Master Menu* may open up to nine separate menu data files, you must load BASIC this way BASIC "MM"/F:9. Since Line 40 sets up key 11, you should name this program MM.BAS. Finally, when returning to *Master Menu* from another program, always return to MENU.1, even if you started from MENU.9.

If anyone has any comments, questions or ideas, write me at 122 Echo Street, Altus AFB, OK 73521, phone: (405) 477-3547.

The listing:

```
10 'MASTER MENU PROGRAM
20 '20 NOV 85   DALE E. BAKER, TSGT, USAF
30 KEY 11,"RUN "+CHR$(34)+"MM"+CHR$(13)    '<CTRL 1> restarts MasterMenu
40 DIM A$(26)
50 DIM B$(26)
60 '
70 '--->Draw Menu
80 '
90 IF N$="" THEN N$="1"
100 IF N=0 THEN N=1
110 GOSUB 610:IF A$(1)="" THEN CLS: PRINT "MENU NOT AVAILABLE.   PRESS ANY KEY T
O CREATE MENU.";N$ ELSE GOTO 140
120 W$=INPUT$(1)
130 GOSUB 1050
140 CLS
150 PRINT CHR$(201);STRING$(78,CHR$(205));CHR$(187);
160 PRINT CHR$(186);STRING$(33," ");"MasterMenu";STRING$(35," ");CHR$(186);
170 LOCATE 2,50:PRINT T$
180 PRINT CHR$(204);STRING$(78,CHR$(205));CHR$(185);
190 FOR X=1 TO 17
200 PRINT CHR$(186);STRING$(78," ");CHR$(186);
210 NEXT
220 PRINT CHR$(200);STRING$(78,CHR$(205));CHR$(188);
230 Y=4:Z=1:FOR X=1 TO 13:LOCATE Y,5:PRINT CHR$(64+Z)+" "+A$(Z):Z=Z+1:Y=Y+1:NEXT
 X
240 Y=4:Z=1:FOR X=1 TO 13:LOCATE Y,45:PRINT CHR$(77+Z)+" "+A$(Z+13):Z=Z+1:Y=Y+1:
NEXT X
250 LOCATE 18,5,0:PRINT "<A-Z> to select Programs                <1-9> to Load a
 Menu"
260 LOCATE 20,5:PRINT   "<ESC> to Quit                           <*> to Create/U
pdate Menu"
270 LOCATE 22,16,0:PRINT "To return to menu after running a program press:"
280 LOCATE 24,37,0:PRINT "<CTRL 1>"
290 '
300 '--->Input Choice
310 '
320 M$=INPUT$(1)
330 IF M$=CHR$(27) THEN SYSTEM
340 IF M$=CHR$(42) THEN GOSUB 450:GOSUB 610:GOSUB 770 ELSE GOTO 370
350 IF C$="Y" THEN GOSUB 1050:GOSUB 770
360 IF C$="N" THEN GOTO 90
```

```
37Ø IF INSTR("123456789",M$) THEN 38Ø ELSE 39Ø
38Ø T=VAL(M$):N=HEX$(T):N$=M$:CLS:GOSUB 61Ø:GOTO 7Ø
39Ø IF INSTR("ABCDEFGHIJKLMNOPQRSTUVWXYZ",M$) THEN 41Ø
4ØØ GOTO 32Ø
41Ø X=INSTR("ABCDEFGHIJKLMNOPQRSTUVWXYZ",M$)
42Ø RUN B$(X)
43Ø GOTO 32Ø
44Ø '
45Ø '--->Display Current Menu
46Ø '
47Ø CLS:LOCATE 2,2Ø:PRINT "Master Menu Edit Routine"
48Ø LOCATE 4,1:PRINT "Current Menu's are:"
49Ø ON ERROR GOTO 57Ø
5ØØ PRINT:PRINT:FILES "A:MENU.?"
51Ø ON ERROR GOTO Ø
52Ø LOCATE 1Ø,28,Ø:PRINT "       Enter Menu #           "
53Ø N$=INPUT$(1)
54Ø IF VAL(N$)=Ø GOTO 53Ø
55Ø N=VAL(N$):LOCATE 1Ø,48,Ø:PRINT N
56Ø GOTO 59Ø
57Ø PRINT "No Menu's On This Disk"
58Ø RESUME NEXT
59Ø RETURN
6ØØ '
61Ø '--->Open Chosen Menu and Input Data
62Ø '
63Ø FOR X=1 TO 26:A$(X)="":NEXT X
64Ø FOR X=1 TO 26:B$(X)="":NEXT X
65Ø T$=""
66Ø M$="MENU."+RIGHT$(N$,1)
67Ø ON ERROR GOTO 74Ø
68Ø OPEN "I",#N,M$
69Ø INPUT #N,T$
7ØØ FOR X=1 TO 26:INPUT #N,A$(X),B$(X):NEXT X
71Ø CLOSE
72Ø FOR X=1 TO 26:IF LEN(A$(X))=Ø THEN A$(X)="":NEXT X
73Ø FOR X=1 TO 26:IF LEN(B$(X))=Ø THEN B$(X)="":NEXT X
74Ø RESUME NEXT
75Ø RETURN
76Ø '
77Ø '--->Display Current Data on Screen
78Ø '
79Ø CLS:LOCATE 2,2Ø,Ø:PRINT "Master Menu Edit Routine"
8ØØ LOCATE 2,5Ø,Ø:PRINT "CURRENT DATA"
81Ø IF T$="" THEN T$="  'EMPTY'"
82Ø PRINT:PRINT"                        Menu Title:  ";T$
83Ø PRINT:PRINT "Description:" TAB(25)"Name:" TAB(45)"Description:" TAB(68)"Name
:"
84Ø Y=8
85Ø FOR X=1 TO 13
86Ø IF A$(X)="" THEN A$(X)="  'EMPTY'"
87Ø IF B$(X)="" THEN B$(X)="  'EMPTY'"
88Ø LOCATE Y,1:PRINT CHR$(64+X)+" "+A$(X) TAB(25) B$(X)
89Ø Y=Y+1
9ØØ NEXT X
91Ø Y=8
92Ø FOR X=14 TO 26
93Ø IF A$(X)="" THEN A$(X)="  'EMPTY'"
94Ø IF B$(X)="" THEN B$(X)="  'EMPTY'"
95Ø LOCATE Y,45:PRINT CHR$(64+X)+" "+A$(X) TAB(68) B$(X)
96Ø Y=Y+1
97Ø NEXT X
98Ø LOCATE 24,2Ø:PRINT "<*> = CHANGE      <ESC> = NO CHANGE"
99Ø W$=INPUT$(1)
1ØØØ IF W$=CHR$(42) THEN C$="Y":GOTO 1Ø3Ø
1Ø1Ø IF W$=CHR$(27) THEN C$="N":GOTO 1Ø3Ø
1Ø2Ø GOTO 99Ø
1Ø3Ø RETURN
1Ø4Ø '
1Ø5Ø '--->Display and Enter/Alter Data
1Ø6Ø '
1Ø7Ø CLS:LOCATE 2,2Ø:PRINT "Master Menu Edit Routine"
1Ø8Ø LOCATE 15,1Ø:PRINT " <CR> = SKIP   <*> = UPDATE    <ESC> = QUIT "
```

```
1090 LOCATE 5,22:PRINT STRING$(23," "):LOCATE 7,22:PRINT STRING$(23," ")
1100 IF T$="" THEN T$="  'EMPTY'"
1110 LOCATE 5,1,0:PRINT "Menu Title:",T$
1120 W$=INPUT$(1)
1130 IF W$=CHR$(13) GOTO 1180
1140 IF W$=CHR$(27) GOTO 1530
1150 IF W$=CHR$(42) LOCATE 6,1:PRINT "Enter New Title": LOCATE 7,1:PRINT STRING$
(20,CHR$(219)):LOCATE 7,1,1:LINE INPUT(20);E$
1160 T$=E$:LOCATE 5,1:PRINT STRING$(45," "):LOCATE 6,1:PRINT STRING$(40," "):LOC
ATE 7,1:PRINT STRING$(40," "):GOTO 1180
1170 GOTO 1120
1180 LOCATE 5,1,0:PRINT "Program Description:  "
1190 LOCATE 7,1,0:PRINT "Program Name[*.BAS]:  "
1200 LOCATE 9,1,0:PRINT "Program Number     :  "
1210 FOR X=1 TO 26
1220 IF A$(X)="" THEN A$(X)="  'EMPTY'"
1230 IF B$(X)="" THEN B$(X)="  'EMPTY'"
1240 LOCATE 5,22,0:PRINT STRING$(23," "):LOCATE 7,22,0:PRINT STRING$(23," ")
1250 LOCATE 5,22,0:PRINT A$(X)
1260 LOCATE 7,22,0:PRINT B$(X)
1270 LOCATE 9,25,0:PRINT X
1280 W$=INPUT$(1)
1290 IF W$=CHR$(13) GOTO 1390
1300 IF W$=CHR$(42) GOTO 1330
1310 IF W$=CHR$(27) THEN X=26:GOTO 1390
1320 GOTO 1280
1330 LOCATE 5,22,0:PRINT STRING$(20,CHR$(219))
1340 LOCATE 7,22,0:PRINT STRING$(12,CHR$(219))
1350 LOCATE 5,22,1:LINE INPUT(20);C$
1360 LOCATE 7,22,1:LINE INPUT(12);D$
1370 A$(X)= C$:B$(X) = D$
1380 LOCATE 5,22,0:PRINT STRING$(23," "):LOCATE 7,22,0:PRINT STRING$(23," ")
1390 NEXT X
1400 '
1410 '--->Delete Old Menu and Store New Menu
1420 '
1430 KILL M$
1440 OPEN "A",#N,M$
1450 PRINT #N,T$
1460 FOR X=1 TO 26
1470 IF A$(X)="  'EMPTY'" THEN A$(X)=""
1480 PRINT #N,A$(X)
1490 IF B$(X)="  'EMPTY'" THEN B$(X)=""
1500 PRINT #N,B$(X)
1510 NEXT X
1520 CLOSE
1530 RETURN
```

Correction

Doctor Up Your Medical Records (April 1986, Page 36): Bill Nieberding advises us that there is a typographical error in the text on Page 37. The two sentences at the end of the fourth paragraph should read as follows:

All characters between { and } should be typed with the GRAPH key enabled. Don't type the { or }.

Correction

Piggybacking (April 1986, Page 20): Bob Jack has passed along two clarifications for this article. First of all, the memory chips should be rated at 120 nanoseconds (or better) access time. Second, in the "Trouble Shooting" section, pins 7, 9, 10 and 11 of the 74LS138 chip are referred to. The article states that pin 7 controls the lowest bank and 11 controls the highest bank. This was accidentally reversed; pin 11 controls the lowest bank and 7 handles the highest.

Software review

Media Master — File Conversion without Wires

Ordinarily, if you wish to transfer a file between two machines of different formats, it is necessary to make the transfer electrically — hook the two machines together via the RS-232 ports, or modems, and send from one to the other. That's cumbersome, slow and not always satisfactory. *Media Master*, by Intersecting Concepts, Inc., allows transfer of files to your Sanyo 555 computer from any of a long list of machines, including the Sanyo CP/M and the Kaypro.

I tested *Media Master* on three different formats: the Kaypro II, Kaypro 10 and Xerox 820; all worked fine. The files were PASCAL source code, text files and *WordStar* files. All were transferred to a standard 360K MS-DOS disk. The PASCAL files were even compiled, without an error, using *Turbo* PASCAL.

BASIC files can also be transferred between machines so long as they have been saved using the ASCII option. Don't expect program files to run, though. The differences between the two dialects must be corrected first. If the program contains PEEKs and POKEs, chances are it will never run correctly unless you figure out exactly what was done in the original version and make it duplicate the same thing on your Sanyo.

Although *Media Master* can be useful for moving source code between machines, its primary use is for file transfer. As an example, the only difference between a Kaypro *WordStar* file and a Sanyo 555 *WordStar* file is the disk format. The Sanyo will not read a Kaypro disk and vice versa. All you need to do is load *Media Master*, tell it one of your drives is a Kaypro and the other an IBM DOS, and copy the Kaypro files to the DOS disk. Then you can call up *WordStar* and edit away. If you need to transfer your work back to the Kaypro disk, just reverse the procedure.

The version of *Media Master* I tested was ported to the 555, but the IBM version should run just fine on the 675, 775 and 885 Sanyo computers. The 555 version runs only on the 555, so if you have both the 555 and one of the 100 percent compatibles, you'll have to decide which to buy.

The documentation is well-done and attractively typeset in a 42-page booklet with a vinyl cover. Operational procedure is explained in a straightforward manner. The manual does assume an IBM PC, so whenever you see IBM, just mentally substitute Sanyo 555.

If you have any need at all to transfer files from one format to another, I don't see how you could go wrong with *Media Master.*

(Intersecting Concepts, Inc., 4573 Heatherglen Ct., Moorpark, CA 93021, 805-529-5073, $39.95)

— Jim Pile

Software review

SmartNotes — The Computerized 'Post-It' Pad

Everywhere I look I see people using those "Post-It" notes by 3M Company, the little (usually yellow) pieces of message paper with adhesive strip that stick to (and peel off of) just about anything. When the history of office communications is written, I suspect Post-It notes will go down with the invention of the QWERTY keyboard and the dictating machine as revolutionary products. Indeed, almost every piece of paper I get in my office has a Post-It note attached.

The trouble is that this only works when you have something onto which the Post-It note can be stuck. And, as one who uses a computer most of the time for writing, note-taking, appointment-making and the like, Post-It notes are just no help.

Enter *SmartNotes* from Personics Corporation, which runs on IBM-compatible systems (but not the 550).

SmartNotes is like Post-It notes for a computer. With this program, you can "attach" notes electronically to just about anything on your computer's screen and, thus, flag it for attention later on.

Take, for example, a letter you have written. It's a letter you can use over again, if you only change a few things here and there — references to a location, a certain product and so on. It is too much trouble to set it up as a mail merge-type document, but you just might mess up if you don't know where all the various pieces of information are buried. *SmartNotes*, which is a memory resident program, is very much like a pad of Post-It notes sitting by your side. All you do is press a key combination and, lo and behold, you can "stick" a note anywhere on the screen. Then, next time you go to that letter, all you have to do is call up the notes you made in the first place and make the changes you need. Presto!

One of the nice things about having a computer with a 20-megabyte hard disk is that there is a lot of storage space. Add a 10-megabyte disk cartridge system, as I have, and you've got a heck of a lot of storage space. You also seem to end up with a lot of files and directories that you don't know anything about! Thanks to *SmartNotes*, you can attach notes to the directories (or anything in DOS for that matter) just the same as you can attach them to, say, letters in an application. Now my computer has a neat notated list of directories that tells me just what is where: no more confusion!

The really nice thing is that you can set up notes with just a couple of keystrokes. You also have a lot of options as well, such as changing the size or color of notes, hiding notes, moving notes around the screen and changing the "context" of a note.

The context is important. *SmartNotes* does not change your application's file, but, instead, writes a separate file for the notes themselves. Each note then is "attached" to a place in your application based on the "context" of the place to which the note is to be pasted. There is a default for this, and you can usually leave it alone. But, as in the aforementioned example, with a DOS directory, the context would be so long that it would cover several lines. If you sorted the directory,

the "context" would be wrong and the note would not appear. So, you edit the context.

You can also page through an application (such as a letter in a word processor) using the page up/down keys and *SmartNotes* only stops when a note appears on the screen. If no notes are found, you are returned directly to your application.

Several utilities are included to list and print notes, open files and the like. Overall, *SmartNotes* has neat and well-thought out design and structure.

There is also a special mode for attaching notes to spreadsheet cells. This is important because, after all, spreadsheets were designed to be recalculated. When you recalculate, however, the "context" changes if you are using the straight text method of attachment used by other applications. Therefore, *SmartNotes* has a different way to do things, and this is a big advantage, too, because it allows you to recalculate, move, reformat and change ranges in a spreadsheet and keep all the notes in the right place.

SmartNotes is memory resident but does not, as far as I have been able to test, conflict with other memory resident programs — about four of which I use together. An important consideration is that it does not seem to care whether it is loaded first, last or in between — making it possible to accede to the requirements of more finicky utilities.

The manual is well-written and uses clear and ample graphics. I recommend this program very highly.

(Personics Corporation, 2352 Main Street, Building Two, Concord, MA 01742, 800-445-3311, $79.95)

— Lonnie Falk

Software review

*Macro*Track* Makes Reliable Economic Predictions

*Macro*Track* is a software package that attempts to predict trends in the general economy. It applies these trends to individual areas of interest. It will predict domestic car sales, interest rates, government surplus and profits, among many others.

*Macro*Track* is simple to use — I sat for hours just watching it plot the future of the automobile industry. After a while, I had to wonder why I would want to know these things anyway, so I started reading all the software's literature. I discovered that *Macro*Track* is targeted more toward the business user than the average home computer hobbyist.

I ran the program periodically and watched the business section of the newspaper much more closely than I ever had before. I found that, in many areas, *Macro*Track* made some amazingly accurate predictions; in a few other areas, it had not done quite so well. It works better in predicting very general trends than specific ones.

I'm no businessman, so in order to give *Macro*Track* a fair shake, I consulted with several friends who are in business. They all agreed *Macro*Track* did as good a job of predicting the general economic trends as any other method they had seen.

One pointed out that a few months is not really enough time to thoroughly evaluate a package such as *Macro*Track*.

I just had to ask the question, "What good is it?" The answer is that knowing the general route the economy is going to follow during the next quarter, or the next year, can be very useful to a businessperson. He needs to have an idea how the economy is likely to move so he can make decisions concerning his own business. He might want to increase or decrease his inventory based on future trends, move his investments into a different area, and so on.

In short, from a business point of view, *Macro*Track* looks like a good buy. It does exactly what it claims to do (not all software does). Black River Software seems to be customer oriented and very helpful when called about one of their products. I called several times and didn't tell them I was reviewing their product. I would assume the polite, helpful comments I received are their usual way of doing business.

As for the software itself, there are a few minor problems I found distracting. Sometimes characters are left on the screen when it appears to me they should be removed. Some operations seem slower than necessary, but I don't know exactly how they are handling the screen, or how many calculations are necessary to get the information to the screen.

I don't consider the minor flaws to be at all serious, and I bow to the good judgment of my business friends who say they think *Macro*Track* is an excellent package. Only hackers notice little dots on the screen anyhow.

(Black River Software, 118 N. Marshall, Suite 150, Winston-Salem, NC 27101, 919-924-6389, $499.95)

— Jim Pile

Software review

Super Batch Takes Monotony out of Frequent Application Use

First you get a fresh cup of coffee then make sure there is plenty of paper for the printer. Next you forward all calls to the secretary, sit down at the computer, turn it on and you're ready for two hours of answering prompts to produce the weekly reports. Once again you think, "There has to be a better way." Merrill Street Software agrees with you. To back this up they have written *Super Batch*, an "MS-DOS Extended Batch File Utility."

The manual supplied with *Super Batch* starts with an overview of what the program can do. Although this part of the book reads like a sales brochure, it gives a good idea of just what *Super Batch* can and will do to make your Sanyo work for you.

Super Batch could be considered a "job control" language because you program it to run a program and pass keystrokes, data or both to the program as it requests it. This is done by creating an ASCII file with MS-DOS commands and keyboard entry in the sequence to run the job.

I did this by setting up a spreadsheet for *CalcStar*. Following the *Super Batch* manual's recommendation, I wrote down each entry as it was made, then built a *Super Batch* file using my

text editor to enter what I did to set up the spreadsheet.

To run it, I typed in SB TESTSC.SB and pressed RETURN; *CalcStar* started, Column A expanded to 20 positions, the headings came up in Row 1 and were promptly written over by the asterisk I wanted displayed on Row 2.

Super Batch feeds an application program one character at a time, including carriage returns. It feeds data so fast that it has a WAIT command to slow the character release down by $^1/_{10}$-second intervals. With this I could see *CalcStar* was receiving carriage returns that were moving the cursor at the wrong time. I found that entering each keystroke on a separate line was causing it to be followed by a carriage return. I changed the batch file from one keystroke per line to ;*;f20<cr> and it worked. The help screen went away, Column A expanded to 20 characters and the cursor stayed in Cell A1.

Once I figured out the difference between keystroke commands and input that requires a carriage return, setting up the batch file for *Super Batch* to run was easy. Watching *CalcStar* execute at the speed that *Super Batch* passes it commands and data made the work in setting it up worthwhile.

During my runs I encountered two problems, one of which required a call to Merrill Street Software. I asked Kenneth VanPelt, author of *Super Batch*, if there was something I was doing wrong since the first ; in my command stream was disappearing, leaving *;f20<cr>. He said because *CalcStar* has a pause between its introduction screen and the startup of the spreadsheet screen, I should try the @S command to fake *CalcStar* into thinking the keyboard buffer was empty. Or, start with an extra ;. The extra ; worked.

The second problem has to do with the Sanyo 55x keyboard. To stop *Super Batch* in the middle of executing, press the ALT and F2 keys. To get the ALT key on a Sanyo, the SHIFT and CTRL keys are pressed at the same time. The combination of SHIFT/CTRL and PF2 on the 55x does not stop *Super Batch*.

"All this is nice but some of my reports require variable entry, like month or dollar amounts," you say. Fear not! *Super Batch* allows variable input by using the @k keyboard entry command. PF keys? Use @e59 through @e68 or @e and a number for any of the extended ASCII codes.

System requirements for *Super Batch* are a Video RAM Board (it won't run under the 55x standard video) and 70K over and above your application's memory requirement. It runs quite well in 256K and works with hard or floppy disks. It is not copy-protected.

The bottom line is that *Super Batch* will save time for anyone who uses an application that requires the same input each time it's run, whether it's daily or weekly reports or testing new programs. The manual is clear and easy to read, and Merrill Street has even included a quick reference card with the package.

(Merrill Street Software, a Division of BCC Inc., 251 Merrill Street, Birmingham, MI 48011, 313-645-5280, $79.95)

— James G. Yearwood

Software review

Opus — A Price and Performance Worth 'Communicating' About

Opus is a communications software package for the Sanyo 550 series. Hardware requirements above the basic system include a serial port and modem. The user must have MS-DOS 2.11 because of the program's extensive use of windows, part-screen scrolling, interrupt, etc. Also, this program includes versions for those users with and without the Video RAM Board.

Opus features are many and varied. This list is a sample of the features:

Auto Logon
Auto Keys
XMODEM protocol (for ASCII and binary transfer)
DC2/DC4 protocol (for ASCII transfer)
Manual ASCII capture (to capture messages, etc.)
Reset Baud (from 110 to 9600)
Reset Data Frame (word length, stop bits, parity)
Reset Duplex (full, half, echo, etc.)
Disk Directory
Type File (to screen or printer)
Copy File (any drive)
Delete, Rename File (any drive)
Change Default Drive
On-Screen Menus (extensive use of this feature)
Windows (for additional menus)
Interrupt driven fully buffered I/O

The set-up procedure is well-detailed and should pose no problems for the novice. If the user has the Video Board in the system, instructions are given for the use of the program *Sidekick*, by Borland International, to be used with *Opus*. I found that in following the instructions, all programs worked as described, including the *Sidekick* option. The procedure is to run *Sidekick*, the *SKGO* patch for the Video Board (supplied), then *Opus*. Also provided is an AUTOEXEC.BAT file to automate the process.

To fully grasp the power of the program and to make full use of its features, it is suggested that the manual be studied carefully. Each of the major functions is described in adequate detail; the novice user should feel very comfortable with this program only after using it one or two times.

It is evident from the package presentation that a substantial amount of time and effort has been placed in this program. The major goal of producing a solid XMODEM transfer has been achieved. In reviewing *Opus*, I was quite impressed.

It is also hard to believe that a program with this many features does not include a comparable price: it is only $20. I highly recommend *Opus* to anyone who needs a communications program for the Sanyo 550 series computers. The price/performance ratio has to be the lowest anywhere!

(Bob Jack Software, 8371 White Road, Burbank, OH 44214, 216-948-2059, $20)

—Edward Champion

Quikpro+II Does the Programming for You

Now there is software that actually writes a program for you. This is the attraction of *Quikpro+II* from ICR FutureSoft. It produces error-free Sanyo BASIC programs for data management and reporting. With a small amount of user work, it writes a specific database input program, prints out a user's manual for that program and writes report-generating programs.

The user can readily comprehend how to operate *Quikpro+II* from the large print 81-page manual, which is both understandable and complete. In using the program, the main options are easily reached from a menu and further work is guided by easy-to-understand statements and questions with simple answers.

The first *Quikpro+II* option guides the user in designing an on-screen entry form for the desired database. As in common database programs, you specify the name of each field of information for storage and its size. A primary key field is assigned for faster data retrieval.

The fields that will only be for numbers may be indicated along with their format. Numeric fields may also be assigned a calculation formula using data entered in other fields, as in a simple spreadsheet. The database is given a name and in about 20 seconds a complete Sanyo BASIC program is written to disk.

This new program can be listed, modified, operated on another computer without the *Quikpro+II* program and even included as the data manipulation portion of another program. As a part-time BASIC programmer, I appreciate how the program code is well-designed, organized and documented with remarks. Data files are written in the fast random file format. In operating the new program, you may add, retrieve, change, delete or search for records. The search function supports searches for a given string of characters that occur in a specified field of any record or anywhere in the database.

Another *Quikpro+II* option generates a Sanyo BASIC program for preparing a report from the database. Two types of reports are possible. The "columnar" report form lists data from specified fields in the database in tabular format. Numeric fields may be summed at the bottom of the page. The "free-form" report form uses data from individual records interspersed with text material. This can be used, for example, to create business letters for multiple mailing or report cards for students. The free-form option operates as a bare-bones line editor.

Sometimes unfilled spaces in database fields result in gaps in a form letter, so care must be taken along the process to reduce this. *Quikpro+II* guides the user in the on-screen preparation of the form, then it takes over and writes the new program. A sorting routine is also written into each program. In operating the new program the user may select the records to include in the report according to the key field, or a search of the database according to a character string or a numeric value.

An extra utility of *Quikpro+II* allows use of databases that were created under a different database program if they are in ASCII.

ICR FutureSoft primarily considers *Quikpro+II* as a database program for non-programmers. It can also be useful for managers or consultants who need to quickly prepare database programs for data entry operators or clients.

Quikpro+II's advantages appear to be its free-form report generator and its production of transportable BASIC programs.

According to ICR FutureSoft, no license fee is required for programs generated by *Quikpro+II* for sale or for use on computers other than the one for which it is licensed. Programmers should note that the manual provides no documentation for the BASIC code placed in the programs that *Quikpro+II* writes.

I was able to go through the entire operation manual in about two hours. After just 30 minutes, I had generated a data entry program for tracking vital data on customers, including a numeric field for a running total of year-to-date orders. Data entry was smooth and fast with my new program. After another 30 minutes I produced a new program to generate a columnar report form of customers. And it was just as easy to create a form letter program.

A minor problem was encountered with an extra blank sheet of paper being advanced for each printed page with my Panasonic KX-P 1091 printer. ICR suggested that I needed to insert alternate printer codes in the generated BASIC program, but they didn't say exactly how to make the correction.

Quikpro+II requires a two-drive Sanyo 555 series computer. It comes as two identical copy-protected diskettes; additional backups may be purchased for $5 each. A 10-day money back guarantee is offered and the company claims only 3 percent of their purchasers ask for a refund. A toll-free order line is available at 1-800-824-7888, operator 441. Telephone support is available from their main office number.

(ICR FutureSoft, 1718 Kingsley Ave., P.O. Box 1446-SA, Orange Park, FL 32073, 904-269-1918, $74.50 plus $4.50 S/H)

— Edward Kerns

You're the Winner when the

A simulation of "Othello," the popular board game, this program requires 128K of memory. The logic used for it was taken from a program I wrote to solve the age-old question of "The Knight's Tour." In this puzzle, the object is to place the knight of a chess set upon any square of the chessboard and move it in such a way that it lands upon each square of the board only once. In fact, the program for this is quite simple and might be fun for you to try.

If you are unfamiliar with the game of "Othello," the object is to end up with more of your chips on the board than those of your opponent, in this case the computer. Each player begins with two circles (chips) of his color on the board.

A move consists of placing one of your chips in such a way as to "trap" the computer's chips and thereby flip them. To trap a chip, place one of your chips so that at least one of the computer's chips is bordered at each end by one of your own chips. This trapping can involve only one chip or several chips.

When you play a chip, all the computer's chips between the one just played and your next chip are flipped. Chips in all directions that are affected by the chip just played may also be flipped. The game is over when there are no more empty squares, or there is no place to put a chip and trap the computer.

Tab Julius, a senior at Castleton State College in Vermont, is director of the school's Academic Computing Center. He will pursue a career in systems programming following graduation. Gary Besaw graduated from Castleton State College in 1985. He is now a programmer/analyst for General Electric in Rutland, Vermont. Gary may be contacted at Box 288A East Street, Clarendon, VT 05759, phone (802) 775-7391. (Please enclose an SASE for a reply when writing.)

Program Description

The program begins on Line 10 by turning off the cursor using the LOCATE command. Lines 20-100 define the colors. This part makes the program much easier to understand. Following this, the program calls the subroutine at 4030. This subroutine is used to write out the titles for the game using the CIRCLE command. This part could be eliminated since it is just for cosmetic purposes.

Lines 120 through 530 initialize all the arrays and variables. Line 540 sends the program out to Line 1710 for the initial setup of play. Line 1710 first calls the subroutine at 3840. This subroutine is used to allow the player to select his color of chip for the game. After getting the color selection, the program calls the board to be generated (lines 590-800) and sets up the beginning board positions of the players' chips.

Line 1770 is the master routine. Depending upon whether you choose black or white, the program uses the routine located at Line 1790 or the routine at 1800. These routines are a WHILE/WEND structure.

Game Play

Your plays are handled in the following manner: First, the program calls up the routines located at lines 1440, 1090 and 1360. These routines are used to display information on the screen such as score, rows, columns and general commands. Next, the program prompts for your column move.

It calls Line 2630 to get the move and, based upon what is entered, it calls various routines. If you enter Q, the program calls Line 1010 to ask if you want to play again. If you enter N, it calls up the routine at 2860. This routine checks through the board to see if there are any moves. If there are moves, it comes back with the message "I see a move, please look again." If there are no moves, it sets the flag PASS equal to true.

If '9' is selected, the game cancels your

Chips are Down

By Gary Besaw and Tab Julius

Variables

TEMPX	Used in selecting a column for the computer's move
TEMPY	Used in selecting a row for the computer move
TEMP1	Used for 'X' position to paint chips on the board
TEMP2	Used for 'Y' postion to paint chips on the board
P	Your score
OP	The computer's score
ANS$	Used for getting answers to prompts
PLAYSCORE	Used to hold your score
OPPSCORE	Used to hold computer's score
PLAYCOL	Your color
TURNCOLOR	The color of the current player
OTHERCOL	The opposite of the turncolor
OPPONCOL	The computer's color
VALID	Flag for a valid move
XSEARCH	Used for searching in the 'X' direction
YSEARCH	Used for searching in the 'Y' direction
XORG	The column location for the move
YORG	The row location for the move
BOARDSPOT	Used for determining the color of the playboard
CHECKX	Increment used in searching in the 'X' direction
CHECKY	Increment used in searching in the 'Y' direction
CHECKBOARD	Used to determine number of flips for computer
T and C	Counters
NOMOVES	Counts the number of times there have been no moves
CHOICE	The desired row and column entries
CHOICE$	The INKEY$ value of the choice
CANCELED	Flag for a canceled move
HOLD	The number of flips for computer for given square
HOLD1	Highest number of flips for computer
PASS	Flag for indicating a valid pass

Arrays

PLAYBOARD	The eight-by-eight game board holding the color value of the chip at that location
SAMEX	Array used to hold all column moves that result in the same number of flips for the compouter
SAMEY	Array used to hold all 'Y' moves that result in the same number of flips for the computer
PRIORITY	The priority of the board positions. Used to help decide on best move
DIRECTION	The array that holds all the possible moves from any spot on the board

move and lets you enter a new one. A number from '1' through '8' is accepted from the INKEY$ by using the VAL option. After getting your column selection, the process is repeated for the row.

Next, the program takes your column and row position, now held in the variables XORG and YORG, and checks to see if it is a valid move. This is checked in the routine located at Line 1920. It works by taking your move position and searching in each of the eight directions to see whether the computer has a chip next to this spot.

If so, it continues going out in that direction (using the routine at Line 2060) counting how many chips of the computer's it finds until it reaches your next chip. If it does not find one of yours at the other end of the chips, it discards that direction and searches the next. If it finds your chip out there, it uses the FOR/NEXT loop at Line 2170 to flip the computer's chips. What it actually does is let the array PLAYBOARD take on your color for those positions where the chips are located.

The program continues flipping chips, if any, in all eight directions until all chips have been flipped. If no chip is flipped, the move is not valid and the flag VALID returns the value not true. If the value is not true, the program beeps (Line 2550) and asks you to enter a new move. If it is a good move, the program calls up the routine at Line 820 to paint the updated board.

The Computer's Move

The computer's moves are controlled by the routine beginning at Line 3160. It begins by searching the board one square at a time to see if any are empty. It does this by using nested FOR/NEXT loops — the outer loop is XORG, the inner loop is YORG.

If it finds an empty square, it begins to search in each of the eight directions to see if one of your chips is next to it. If one is, it calls the routine at Line 3420. This routine continues out in the same direction as the chip it found to see if it can flip any chips. If it can, it counts the number of chips that will be flipped by that move in all directions.

In lines 3320-3340, it compares the number of chips that can be flipped there to the number of chips that can be flipped by the last move. If more chips can be flipped here, then the holders TEMPX and TEMPY take on the value of XORG and YORG.

It continues until it has searched the whole board and found the "best" move. It considers the corners to be the best moves, so if it can get a corner (PRIORITY=1), it takes it no matter how many chips it flips. If several of the moves result in the same number of chips, then it places them in the arrays SAMEX and SAMEY and randomly chooses a move.

Ending

If the computer finds it has no move, it tells you and lets you go again. If you pass and it passes (Line 3700), or if it determines that all the squares are occupied (Line 990), it then ends the game and determines the winner.

The listing:

```
1Ø LOCATE 1Ø,1Ø,Ø
2Ø REM DEFINE COLORS
3Ø BLACK =Ø
4Ø BLUE  =1
5Ø GREEN =2
6Ø LGHTBLUE=3
7Ø RED=4
8Ø PURPLE=5
9Ø YELLOW=6
1ØØ WHITE=7
11Ø GOSUB 4Ø3Ø
12Ø TEMPX=2:TEMPY=2
13Ø DIM PLAYBOARD(9,9),SAMEX(64),SAMEY(6
4),PRIORITY(9,9),DIRECTION(9,2)
14Ø COLOR BLACK,BLUE
15Ø FOR X=1 TO 8
16Ø FOR Y=1 TO 8
17Ø PLAYBOARD(X,Y)=GREEN
18Ø NEXT Y
19Ø NEXT X
2ØØ PLAYBOARD(4,5)=BLACK
21Ø PLAYBOARD(5,4)=BLACK
22Ø PLAYBOARD(4,4)=WHITE
23Ø PLAYBOARD(5,5)=WHITE
24Ø REM DATA FOR PRIORITY
25Ø DATA 1,8,2,3,3,2,8,1
26Ø DATA 8,9,7,6,6,7,9,8
27Ø DATA 2,7,4,5,5,4,7,2
28Ø DATA 3,6,5,Ø,Ø,5,6,3
29Ø DATA 3,6,5,Ø,Ø,5,6,3
3ØØ DATA 2,7,4,5,5,4,7,2
31Ø DATA 8,9,7,6,6,7,9,8
32Ø DATA 1,8,2,3,3,2,8,1
33Ø REM LOAD PRIORITIES
34Ø FOR X=1 TO 8
35Ø FOR Y=1 TO 8
36Ø READ PRIORITY(X,Y)
37Ø NEXT Y
38Ø NEXT X
39Ø REM DEFINE X AND Y OFFSETS
4ØØ DATA 1,Ø,1,-1,Ø,-1,-1,-1,-1,Ø,-1,1,Ø
,1,1,1
48Ø FOR X=1 TO 8
49Ø FOR Y=1 TO 2
5ØØ READ DIRECTION(X,Y)
51Ø NEXT Y
52Ø NEXT X
53Ø TRUE=1:FALSE=NOT TRUE
54Ø GOTO 171Ø
55Ø REM INITIALIZE SCREEN
56Ø CLS
57Ø COLOR BLUE,BLUE
58Ø LINE(Ø,Ø)-(639,224),,BF
59Ø COLOR GREEN
6ØØ LINE(221,21)-(545,181),,BF
61Ø COLOR BLACK
62Ø LINE(217,21)-(545,185),,B
63Ø LINE(218,18)-(544,184),,B
64Ø LINE(219,19)-(543,183),,B
65Ø LINE(22Ø,2Ø)-(542,182),,B
66Ø LINE(221,21)-(541,181),,B
67Ø LINE(221,41)-(541,41)
68Ø LINE(221,61)-(541,61)
69Ø LINE(221,81)-(541,81)
7ØØ LINE(221,1Ø1)-(541,1Ø1)
71Ø LINE(221,121)-(541,121)
72Ø LINE(221,141)-(541,141)
73Ø LINE(221,161)-(541,161)
74Ø LINE(261,21)-(261,181)
75Ø LINE(3Ø1,21)-(3Ø1,181)
76Ø LINE(341,21)-(341,181)
77Ø LINE(381,21)-(381,181)
78Ø LINE(421,21)-(421,181)
79Ø LINE(461,21)-(461,181)
8ØØ LINE(5Ø1,21)-(5Ø1,181)
81Ø RETURN
82Ø REM UPDATE BOARD
83Ø FOR X1=1 TO 8
84Ø FOR Y1=1 TO 8
85Ø IF PLAYBOARD(X1,Y1)=GREEN THEN 92Ø
86Ø LET TEMP1=(X1*4Ø)+2ØØ:TEMP2=(Y1*2Ø)+
11
87Ø IF PLAYBOARD(X1,Y1)=PLAYCOL THEN P=P
+1
88Ø IF PLAYBOARD(X1,Y1)=OPPONCOL THEN OP
=OP+1
89Ø COLOR PLAYBOARD(X1,Y1)
9ØØ CIRCLE(TEMP1,TEMP2),14
91Ø PAINT (TEMP1,TEMP2)
92Ø NEXT Y1
93Ø NEXT X1
95Ø LINE (Ø,3Ø)-(185,5Ø),RED,BF
96Ø COLOR BLACK,RED
97Ø LOCATE 5,1:PRINT"YOUR SCORE IS NOW:
";P
```

```
98Ø LOCATE 6,1:PRINT"  MY SCORE IS NOW:
";OP
99Ø IF P+OP=64 THEN 1ØØØ ELSE 1Ø65
1ØØØ LINE (Ø,13Ø)-(18Ø,1ØØ),RED,BF:LOCAT
E 14,2:PRINT "THE GAME IS OVER"
1ØØ5 LOCATE 15,2:IF P>OP THEN PRINT "YOU
 HAVE WON!" ELSE PRINT "I HAVE WON"
1Ø1Ø FOR X=1 TO 3ØØØ:NEXT X:CLS:LOCATE 1
2,1Ø:PRINT"DO YOU WISH TO PLAY AGAIN?"
1Ø15 LOCATE 13,11:PRINT "Y FOR YES, N FO
R NO"
1Ø2Ø WHILE 1=1
1Ø3Ø ANS$=INKEY$
1Ø4Ø IF ANS$="" THEN WEND
1Ø5Ø IF ANS$="Y" THEN GOTO 171Ø
1Ø6Ø IF ANS$="N" THEN CLS:END
1Ø65 PLAYSCORE=P:OPPSCORE=OP
1Ø7Ø OP=Ø:P=Ø
1Ø8Ø RETURN
1Ø9Ø COLOR BLACK
11ØØ FOR X=Ø TO 336 STEP 48:LINE(X,214)-
(X+36,224),,BF:NEXT X
111Ø COLOR GREEN
112Ø FOR X=Ø TO 336 STEP 48:LINE(X,214)-
(X+36,224),,B:NEXT X
113Ø COLOR GREEN,BLACK
114Ø LOCATE 24,31:PRINT"1";
115Ø LOCATE 24,36:PRINT"2";
116Ø LOCATE 24,41:PRINT"3";
117Ø LOCATE 24,46:PRINT"4";
118Ø LOCATE 24,51:PRINT"5";
119Ø LOCATE 24,56:PRINT"6";
12ØØ LOCATE 24,61:PRINT"7";
121Ø LOCATE 24,66:PRINT"8";
122Ø FOR X=Ø TO 336 STEP 48:LINE(X,214)-
(X+36,224),,B:NEXT X
123Ø RETURN
124Ø COLOR BLACK
125Ø LINE(528,214)-(528+36,224),,BF
126Ø COLOR GREEN
127Ø LINE(528,214)-(528+36,224),,B
128Ø COLOR GREEN,BLACK
129Ø LOCATE 24,75:PRINT "Q";
13ØØ LINE(528,214)-(528+36,224),,B
131Ø LOCATE 23,73:PRINT "QUIT";
132Ø LINE (557,2Ø6)-(545,214),BLACK
133Ø LINE (558,2Ø6)-(546,214),BLACK
134Ø LINE (559,2Ø6)-(547,214),BLACK
135Ø RETURN
136Ø BACKGROUND=BLACK:LETTERS=GREEN
137Ø LOCATE 25,1:PRINT "KEYS 1-8 FOR COL
UMN/ROW MOVES";
138Ø LOCATE 25,31:PRINT "N IF THERE IS N
O MOVE";
139Ø LOCATE 25,53:PRINT "Q EXITS GAME";
14ØØ LOCATE 25,66:PRINT"9 CANCELS MOVE";
141Ø LINE (424,2Ø6)-(4ØØ,214),BLACK:LINE
(425,2Ø6)-(4Ø1,214),BLACK
142Ø LINE (426,2Ø6)-(4Ø2,214),BLACK
143Ø RETURN
144Ø COLOR GREEN,BLACK
145Ø LOCATE 4,25:PRINT "1>":LOCATE 4,7Ø:
PRINT"<1"
146Ø LOCATE 7,25:PRINT "2>":LOCATE 7,7Ø:
PRINT"<2"
147Ø LOCATE 1Ø,25:PRINT "3>":LOCATE 1Ø,7
Ø:PRINT"<3"
148Ø LOCATE 12,25:PRINT "4>":LOCATE 12,7
Ø:PRINT"<4"
149Ø LOCATE 15,25:PRINT "5>":LOCATE 15,7
Ø:PRINT"<5"
15ØØ LOCATE 17,25:PRINT "6>":LOCATE 17,7
Ø:PRINT"<6"
151Ø LOCATE 19,25:PRINT "7>":LOCATE 19,7
Ø:PRINT"<7"
152Ø LOCATE 22,25:PRINT "8>":LOCATE 22,7
Ø:PRINT"<8"
153Ø COLOR GREEN,BLACK
154Ø X=Ø
155Ø FOR Y=3Ø TO 65 STEP 5
156Ø X=X+1
157Ø LOCATE 2,Y
158Ø PRINT X
159Ø NEXT Y
16ØØ RETURN
161Ø REM CLEAR BOTTOM OF BOARD
162Ø LINE (Ø,191)-(64Ø,185),BLUE,BF
163Ø RETURN
164Ø REM REMOVE SIDE NUMBERS
165Ø LINE (19Ø,2Ø)-(21Ø,178),BLUE,BF
166Ø LINE (55Ø,2Ø)-(57Ø,178),BLUE,BF
167Ø RETURN
168Ø REM ROUTINE TO REMOVE OTHER NUMBERS
169Ø LINE (22Ø,Ø)-(6ØØ,16),BLUE,BF
17ØØ RETURN
171Ø REM INITIALIZE PLAY
172Ø GOSUB 384Ø
173Ø CLS
174Ø LINE(Ø,Ø)-(639,199),BLUE,BF
175Ø GOSUB 59Ø
176Ø GOSUB 82Ø
177Ø REM MASTER ROUTINE
178Ø IF PLAYCOL=BLACK THEN 18ØØ ELSE 179
Ø
179Ø TURNCOLOR=PLAYCOL:OTHERCOL=OPPONCOL
:WHILE 1=1:GOSUB 188Ø:GOSUB 316Ø:
                  GOSUB 188Ø:GOSUB 228Ø:
WEND
18ØØ TURNCOLOR=PLAYCOL
181Ø OTHERCOLOR=OPPONCOL
182Ø WHILE 1=1
183Ø GOSUB 228Ø
184Ø GOSUB 188Ø
185Ø GOSUB 316Ø
186Ø GOSUB 188Ø
187Ø WEND
188Ø REM PLAYER MOVES
189Ø OTHERCOLOR=TURNCOLOR
19ØØ IF TURNCOLOR=PLAYCOL THEN TURNCOLOR
=OPPONCOL ELSE TURNCOLOR=PLAYCOL
191Ø RETURN
192Ø REM CHECK FOR VALID MOVE
1925 VALID=Ø
193Ø IF PLAYBOARD(XORG,YORG)<>GREEN THEN
 RETURN
194Ø VALID=Ø
195Ø FOR X=1 TO 8
196Ø XSEARCH=XORG+DIRECTION(X,1)
197Ø YSEARCH=YORG+DIRECTION(X,2)
198Ø IF XSEARCH<1 OR XSEARCH>8 THEN 2Ø4Ø
199Ø IF YSEARCH<1 OR YSEARCH>8 THEN 2Ø4Ø
2ØØØ BOARDSPOT=PLAYBOARD(XSEARCH,YSEARCH
)
2Ø1Ø IF BOARDSPOT=GREEN THEN GOTO 2Ø4Ø
2Ø2Ø IF BOARDSPOT=TURNCOLOR THEN GOTO 2Ø
4Ø
2Ø3Ø IF BOARDSPOT=OTHERCOLOR THEN GOSUB
2Ø6Ø
2Ø4Ø NEXT X
2Ø5Ø RETURN
2Ø6Ø T=1
2Ø7Ø CHECKX=XORG:CHECKY=YORG
2Ø8Ø FOR W=1 TO 7
2Ø9Ø CHECKX=CHECKX+DIRECTION(X,1)
21ØØ CHECKY=CHECKY+DIRECTION(X,2)
211Ø IF CHECKX<1 OR CHECKX>8 THEN GOTO 2
25Ø
212Ø IF CHECKY<1 OR CHECKY>8 THEN GOTO 2
25Ø
213Ø CHECKBOARD=PLAYBOARD(CHECKX,CHECKY)
214Ø IF CHECKBOARD=TURNCOLOR THEN VALID=
1:GOTO 217Ø
215Ø IF CHECKBOARD=GREEN THEN T=Ø: RETUR
N
216Ø IF CHECKBOARD=OTHERCOLOR THEN T=T+1
:GOTO 225Ø
```

```
217Ø FOR COUNT =T+1 TO 1 STEP-1
218Ø LET PLAYBOARD(CHECKX,CHECKY)=TURNCO
LOR:VALID=1
219Ø CHECKX=CHECKX-DIRECTION(X,1)
22ØØ CHECKY=CHECKY-DIRECTION(X,2)
221Ø NEXT COUNT
222Ø VALID=1
223Ø T=Ø
224Ø RETURN
225Ø NEXT W
226Ø T=Ø
227Ø RETURN
228Ø GOSUB 144Ø
229Ø GOSUB 1Ø9Ø
23ØØ GOSUB 136Ø
231Ø COLOR WHITE,BLUE
232Ø LOCATE 23,2
233Ø PRINT "PLEASE SELECT YOUR COLUMN"
234Ø GOSUB 263Ø
235Ø IF NOMOVES>Ø THEN RETURN
236Ø XORG=CHOICE
237Ø COLOR BLUE,BLUE
238Ø LINE (Ø,185)-(2Ø8,175),BLUE,BF
239Ø GOSUB 161Ø
24ØØ GOSUB 153Ø
241Ø GOSUB 136Ø
242Ø COLOR WHITE,BLUE
243Ø LOCATE 21,2:PRINT "COLUMN";CHOICE;
244Ø LOCATE 23,2
245Ø PRINT"PLEASE SELECT YOUR ROW"
246Ø GOSUB 263Ø
2465 IF CANCELLED=TRUE THEN XORG=Ø:YORG=
Ø:GOTO 228Ø
247Ø COLOR WHITE,BLUE:LOCATE 22,2:PRINT
"ROW";CHOICE;
248Ø LOCATE 23,2
249Ø COLOR BLUE,BLUE
25ØØ LINE (Ø,185)-(183,175),BLUE,BF
251Ø YORG=CHOICE
252Ø GOSUB 164Ø
253Ø GOSUB 168Ø
254Ø GOSUB 192Ø
255Ø IF VALID=1 THEN GOTO 261Ø ELSE 256Ø
256Ø FOR DELAY= 1 TO 5:BEEP:NEXT DELAY
257Ø COLOR WHITE,BLUE:LOCATE 23,2:PRINT
"SORRY INVALID MOVE"
258Ø FOR X=1 TO 3ØØØ:NEXT X
259Ø LINE (Ø,185)-(21Ø,175),BLUE,BF
26ØØ GOSUB 144Ø:GOSUB 1Ø9Ø:GOSUB 136Ø:GO
TO 231Ø
261Ø GOSUB 82Ø
262Ø RETURN
263Ø REM GET MOVES
264Ø WHILE 1=1
265Ø CHOICE$=INKEY$
266Ø IF CHOICE$="" THEN WEND
267Ø IF CHOICE$="Q" THEN GOTO 1Ø1Ø
268Ø IF CHOICE$="N" THEN GOSUB 286Ø ELSE
 278Ø
269Ø LINE (Ø,13Ø)-(185,1ØØ),RED,BF
27ØØ IF PASS<>TRUE THEN GOTO 271Ø ELSE 2
74Ø
271Ø COLOR WHITE,RED:LOCATE 14,1:PRINT"P
LEASE LOOK AGAIN"
272Ø LOCATE 15,1:PRINT "I SEE A MOVE"
273Ø FOR X=1 TO 2ØØØ:NEXT X:LINE(Ø,13Ø)-
(185,1ØØ),BLUE,BF
     :COLOR WHITE,BLUE:LOCATE 23,2:PRINT
"PLEASE SELECT YOUR COLUMN":GOTO 264Ø
274Ø NOMOVES=NOMOVES+1
275Ø COLOR WHITE,RED:LOCATE 14,1:PRINT "
SINCE YOU HAVE NO MOVES"
276Ø LOCATE 15,3:PRINT"I WILL GO AGAIN":
FOR X=1 TO 3ØØØ:NEXT X
277Ø LINE(Ø,13Ø)-(185,1ØØ),BLUE,BF:RETUR
N
278Ø NOMOVES=Ø:CHOICE=VAL(CHOICE$)
279Ø IF CHOICE<1 OR CHOICE> 9 THEN BEEP:
GOTO 264Ø
28ØØ IF CHOICE=9 THEN CANCELLED=TRUE:GOT
O 281Ø
2805 CANCELLED=FALSE:GOTO 285Ø
281Ø LINE(Ø,185)-(21Ø,175),BLUE,BF
282Ø LOCATE 23,2 :PRINT "CANCELLED; ENTE
R CHANGE"
283Ø FOR X=1 TO 175Ø:NEXT X
285Ø RETURN
286Ø REM CHECK FOR NO MOVE
287Ø PASS=TRUE
288Ø LINE(Ø,185)-(21Ø,175),BLUE,BF
289Ø LOCATE 23,2:PRINT "LET ME CHECK THA
T"
29ØØ FOR XORG=1 TO 8
291Ø FOR YORG=1 TO 8
292Ø IF PLAYBOARD(XORG,YORG)<>GREEN THEN
 3Ø3Ø
293Ø FOR X=1 TO 8
294Ø XSEARCH=XORG+DIRECTION(X,1):YSEARCH
=YORG+DIRECTION(X,2)
295Ø IF XSEARCH<1 OR XSEARCH>8 THEN GOTO
 3Ø2Ø
296Ø IF YSEARCH<1 OR YSEARCH>8 THEN GOTO
 3Ø2Ø
297Ø BOARDSPOT=PLAYBOARD(XSEARCH,YSEARCH
)
298Ø IF BOARDSPOT=GREEN THEN 3Ø2Ø
299Ø IF BOARDSPOT=TURNCOLOR THEN 3Ø2Ø
3ØØØ IF BOARDSPOT=OTHERCOLOR THEN GOSUB
3Ø6Ø
3Ø1Ø IF PASS=FALSE THEN RETURN
3Ø2Ø NEXT X
3Ø3Ø NEXT YORG
3Ø4Ø NEXT XORG
3Ø5Ø RETURN
3Ø6Ø CHECKX=XORG:CHECKY=YORG
3Ø7Ø FOR W=1 TO 7
3Ø75 CHECKX=CHECKX+DIRECTION(X,1):CHECKY
=CHECKY+DIRECTION(X,2)
3Ø8Ø IF CHECKX<1 OR CHECKX>8 THEN GOTO 3
14Ø
3Ø9Ø IF CHECKY<1 OR CHECKY>8 THEN GOTO 3
14Ø
31ØØ CHECKBOARD=PLAYBOARD(CHECKX,CHECKY)
311Ø IF CHECKBOARD=GREEN THEN RETURN
312Ø IF CHECKBOARD=OTHERCOLOR THEN GOTO
314Ø
313Ø IF CHECKBOARD=TURNCOLOR THEN PASS=F
ALSE:RETURN
314Ø NEXT W
315Ø RETURN
316Ø REM COMPUTER MOVE
317Ø LINE(Ø,185)-(2Ø8,175),BLUE,BF
318Ø COLOR WHITE,BLUE:LOCATE 23,2:PRINT
"THINKING...PLEASE WAIT"
319Ø FOR XORG=1 TO 8
32ØØ FOR YORG=1 TO 8
321Ø IF PLAYBOARD(XORG,YORG)<>GREEN THEN
 336Ø
322Ø FOR X=1 TO 8
323Ø XSEARCH=XORG+DIRECTION(X,1)
324Ø YSEARCH=YORG+DIRECTION(X,2)
325Ø IF XSEARCH<1 OR XSEARCH>8 THEN GOTO
 331Ø
326Ø IF YSEARCH<1 OR YSEARCH>8 THEN GOTO
 331Ø
327Ø BOARDSPOT=PLAYBOARD(XSEARCH,YSEARCH
)
328Ø IF BOARDSPOT=GREEN THEN GOTO 331Ø
329Ø IF BOARDSPOT=TURNCOLOR THEN GOTO 33
1Ø
33ØØ IF BOARDSPOT=OTHERCOLOR THEN GOSUB
342Ø
331Ø NEXT X
332Ø IF HOLD>Ø AND PRIORITY(XORG,YORG)=1
 THEN HOLD1=HOLD:TEMPX=XORG:
                    TEMPY=YORG:HOLD=Ø:NOMOVE
```

```
S=Ø:GOSUB 354Ø:RETURN
333Ø IF HOLD=HOLD1 AND BOARDPRIORITY<>9
AND HOLD<>Ø THEN J=J+1:SAMEX(J)=TEMPX:
                              SAMEY(J)=TEMPY
:TEMPX=XORG:TEMPY=YORG:NOMOVES=Ø
334Ø IF HOLD>HOLD1 THEN HOLD1=HOLD:TEMPX
=XORG:TEMPY=YORG:NOMOVES=Ø:J=Ø
335Ø HOLD=Ø
336Ø NEXT YORG
337Ø NEXT XORG
338Ø IF J>Ø THEN J=J+1:SAMEX(J)=TEMPX:SA
MEY(J)=TEMPY:GOSUB 444Ø
339Ø IF HOLD1=Ø THEN TEMPX=Ø:TEMPY=Ø:GOS
UB 37ØØ
34ØØ GOSUB 354Ø
341Ø RETURN
342Ø C=1
343Ø CHECKX=XORG:CHECKY=YORG
344Ø FOR W=1 TO 6
345Ø CHECKX=CHECKX+DIRECTION(X,1):CHECKY
=CHECKY+DIRECTION(X,2)
346Ø IF CHECKX<1 OR CHECKX>8 THEN GOTO 3
52Ø
347Ø IF CHECKY<1 OR CHECKY>8 THEN GOTO 3
52Ø
348Ø CHECKBOARD=PLAYBOARD(CHECKX,CHECKY)
349Ø IF CHECKBOARD=GREEN THEN C=Ø:RETURN
35ØØ IF CHECKBOARD=OTHERCOLOR THEN C=C+1
 :GOTO 352Ø
351Ø HOLD=HOLD+C:RETURN
352Ø NEXT W
353Ø RETURN
354Ø FOR X=1 TO 8
355Ø LET PLAYBOARD(TEMPX,TEMPY)=TURNCOLO
R
356Ø COLOR WHITE,BLUE
357Ø LOCATE 21,2:PRINT "COLUMN";TEMPX:LO
CATE 22,2:PRINT "ROW";TEMPY
3575 IF TEMPX=Ø AND TEMPY=Ø THEN RETURN
358Ø LET XORG=TEMPX:YORG=TEMPY
359Ø XSEARCH=TEMPX+DIRECTION(X,1):YSEARC
H=TEMPY+DIRECTION(X,2)
36ØØ IF XSEARCH<1 OR XSEARCH>8 THEN 366Ø
361Ø IF YSEARCH<1 OR YSEARCH>8 THEN 366Ø
362Ø BOARDSPOT=PLAYBOARD(XSEARCH,YSEARCH
)
363Ø IF BOARDSPOT=GREEN THEN GOTO 366Ø
364Ø IF BOARDSPOT=TURNCOLOR THEN GOTO 36
6Ø
365Ø IF BOARDSPOT=OTHERCOLOR THEN GOSUB
2Ø6Ø
366Ø NEXT X
367Ø GOSUB 82Ø
368Ø HOLD1=Ø
369Ø RETURN
37ØØ COLOR WHITE,RED:LINE (Ø,13Ø)-(18Ø,1
ØØ),RED,BF
371Ø IF NOMOVE=Ø THEN GOTO 379Ø ELSE 372
Ø
372Ø LINE(Ø,13Ø)-(18Ø,1ØØ),RED,BF
373Ø LOCATE 14,2:PRINT "THERE ARE NO MOV
ES"
374Ø LOCATE 15,2:PRINT "SO THE GAME IS O
VER"
375Ø FOR X=1 TO 3ØØØ:NEXT X:LINE(Ø,13Ø)-
(18Ø,1ØØ),BLUE,BF
376Ø IF PLAYSCORE>OPPSCORE THEN 377Ø ELS
E 378Ø
377Ø LINE(Ø,13Ø)-(18Ø,1ØØ),RED,BF:LOCATE
 14,2:PRINT "YOU ARE THE WINNER":
                          GOTO 1Ø1Ø
378Ø LINE(Ø,13Ø)-(18Ø,1ØØ),RED,BF:LOCATE
 14,2:PRINT "I HAVE WON":GOTO 1Ø1Ø
379Ø LINE(Ø,13Ø)-(18Ø,1ØØ),RED,BF
38ØØ LOCATE 14,2:PRINT "I HAVE NO MOVE":
LOCATE 15,3:PRINT"GO AGAIN"
381Ø FOR X=1 TO 2ØØØ:NEXT X:LINE(Ø,13Ø)-
(18Ø,1ØØ),BLUE,BF
382Ø NOMOVE=NOMOVE+1
383Ø RETURN
384Ø LINE(Ø,199)-(639,Ø),GREEN,BF
385Ø COLOR WHITE,GREEN
386Ø CIRCLE (17Ø,95),6Ø
387Ø PAINT(17Ø,95)
388Ø COLOR BLACK,GREEN
389Ø CIRCLE (35Ø,95),6Ø
39ØØ PAINT(35Ø,95)
391Ø LINE(4Ø,155)-(55Ø,18Ø),RED,BF
392Ø SYMBOL(8Ø,16Ø),"PLEASE SELECT THE F
IRST LETTER OF THE COLOR THAT YOU WANT",
1,1,WHITE
393Ø SYMBOL(23Ø,17Ø),"BLACK PLAYS FIRST"
,1,1,BLACK
394Ø SYMBOL(16Ø,95),"W",3,3,BLACK
395Ø SYMBOL(34Ø,95),"B",3,3,WHITE
396Ø WHILE 1=1
397Ø CHOICE$=INKEY$
398Ø IF CHOICE$="" THEN WEND
399Ø IF CHOICE$="B" THEN PLAYCOL=BLACK:O
PPONCOL=WHITE:GOTO 4Ø2Ø
4ØØØ IF CHOICE$="W" THEN PLAYCOL=WHITE:O
PPONCOL=BLACK:GOTO 4Ø2Ø
4Ø1Ø IF CHOICE$<>"B" OR CHOICE$<>"W" THE
N BEEP:GOTO 392Ø
4Ø2Ø RETURN
4Ø3Ø COLOR BLUE,BLUE
4Ø4Ø LINE (Ø,199)-(639,Ø),,BF
4Ø5Ø COLOR LGHTBLUE
4Ø6Ø FOR X=67 TO 7Ø
4Ø7Ø CIRCLE(14Ø,13Ø),X,,,3
4Ø8Ø NEXT X
4Ø9Ø REM T
41ØØ FOR X=5Ø TO 52
411Ø CIRCLE(215,12Ø),X,.24,.76,2
412Ø NEXT X
413Ø FOR X=5Ø TO 52
414Ø CIRCLE(19Ø,8Ø),X,.1,.33,.9
415Ø NEXT X
416Ø REM H
417Ø FOR X=43 TO 45
418Ø CIRCLE (27Ø,118),X,.24,.76,2
419Ø NEXT X
42ØØ FOR X= 45 TO 47
421Ø CIRCLE(27Ø,16Ø),X,.65,Ø,2
422Ø NEXT X
423Ø REM E
424Ø FOR X=3Ø TO 32
425Ø CIRCLE (345,127),X,.24,Ø,2
426Ø NEXT X
427Ø FOR X=44 TO 46
428Ø CIRCLE(345,9Ø),X,.21,.29,2
429Ø NEXT X
43ØØ FOR X=48 TO 5Ø
431Ø CIRCLE(415,9Ø),X,.24,.76,2
432Ø NEXT X
433Ø FOR X=48 TO 5Ø
434Ø CIRCLE (47Ø,85),X,.24,.76,2
435Ø NEXT X
436Ø FOR X=25 TO 27
437Ø CIRCLE (52Ø,1ØØ),X,,,1
438Ø NEXT X
439Ø LINE(8Ø,2Ø5)-(565,15Ø)
44ØØ LINE(8Ø,2Ø6)-(565,151)
441Ø LOCATE 1,4:PRINT" A COMPUTER GAME F
OR THE SANYO MBC55Ø/555
442Ø FOR X= 1 TO 5ØØØ:NEXT X
443Ø RETURN
444Ø LET MOVE=INT(RND*J)+1
445Ø TEMPX=SAMEX(MOVE):TEMPY=SAMEY(MOVE)
446Ø FOR Q=1 TO J
447Ø LET SAMEX(Q)=Ø:SAMEY(Q)=Ø
448Ø NEXT Q
449Ø J=Ø
45ØØ RETURN
451Ø REM H
```

Sanyo Synthesizer is a fast, easy way to play and store tunes without the hassle of setting and resetting durations, pitches and rests. You don't enter a tune one note at a time, but actually play it right on the keyboard.

The program makes use of the sound driver published in the December 1984 issue of SOFT SECTOR (*Run'Round*, Page 28) to produce musical notes. You can play, save, load and compile musical tunes (more on this later).

Type in and save Listing 1. When the program is run, the title appears on the screen with a graphics representation of the Sanyo's keyboard, along with the keys that are used labeled with their note values. The keys are arranged like a piano keyboard, using the second and bottom rows of the keyboard for the naturals and the top and third rows for the sharps and flats. This provides a total of four octaves. A menu and other controls of the program are also displayed.

There are three different modes. The mode you are currently in is displayed at the bottom of the screen. When the program begins, you are in the PLAY mode. Pressing any of the keys displayed on the graphics keyboard will sound that note. To stop the note, press the space bar, otherwise that same note keeps on playing until another note is played or you press the space bar, RETURN or ESC.

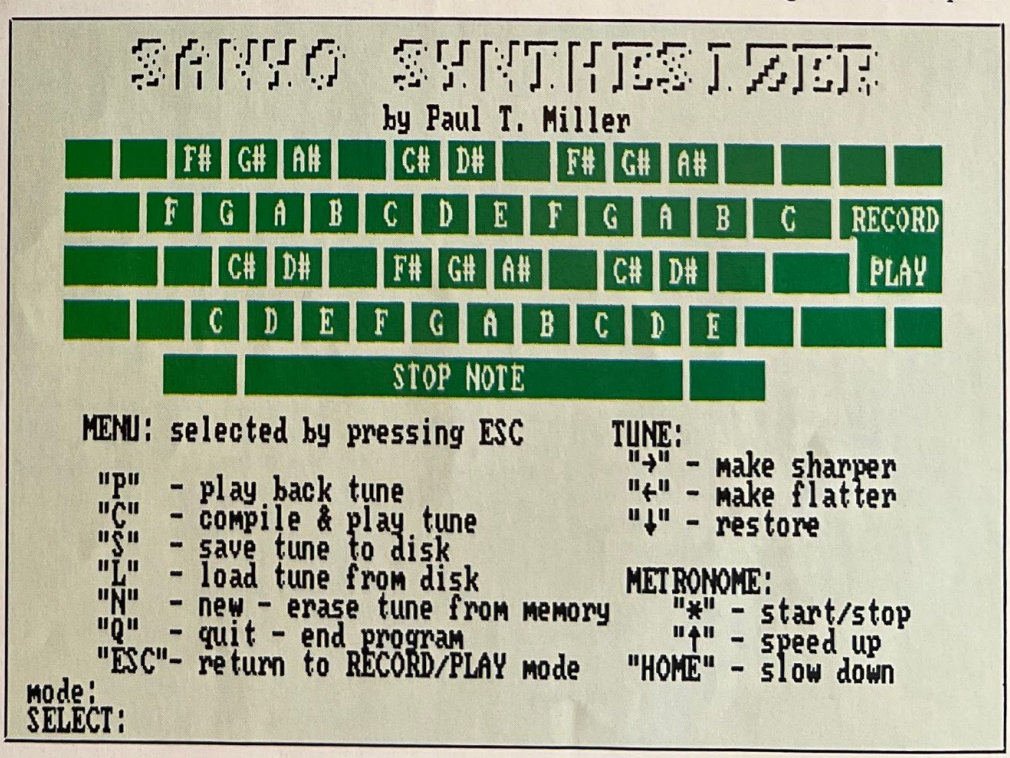
When a note is playing, there is a "pulsing" sound. This is because the program is actually toggling the note many times per second. This happens so the program can automatically save notes and rest durations.

Pressing RETURN toggles you in and out of the RECORD mode. Pressing keys in this mode saves the note values to the P array for later playing and saving. Pressing RETURN again returns you to the PLAY mode.

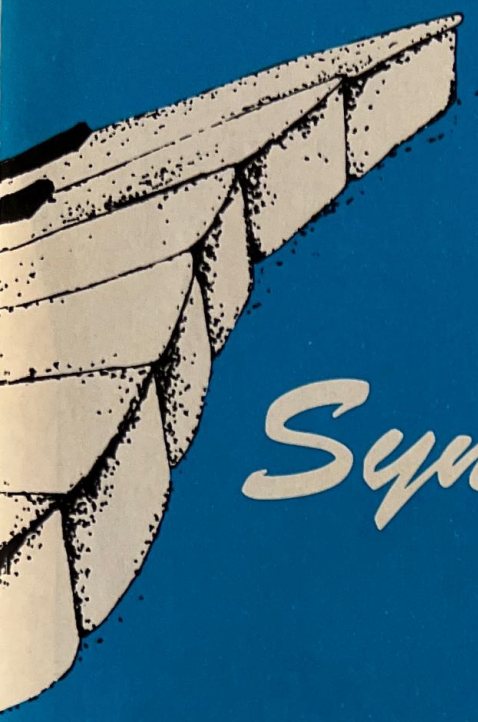
Pressing the ESCape key in either mode transfers you to the SELECT mode, which prompts you to select an option from the on-screen menu. Pressing ESCape there returns you to the PLAY mode.

Your menu options are: play back tune, compile and play tune, save tune to disk, load tune from disk, new — erase tune from memory, and quit — end program. Pressing 'P' plays back a tune (if it has been recorded) exactly as you played it — even with the pulses. Pressing 'S' asks you to enter a filename so the tune can be saved to disk. Pressing 'L' prompts for a filename to load a tune from the disk. Pressing 'N' erases the tune from memory so you can start over. Pressing 'Q' ends the program. And finally, pressing 'C' starts "compiling" your tune. It counts the number of pulses for each note and rest, then stores those values into the CO array. When it is finished, it plays back the tune in pure, steady notes without the pulses.

If your perception of musical notes is sometimes off, or you feel the notes don't sound right, just TUNE it! While in the PLAY or RECORD mode, pressing the up and down arrows changes the note's pitch



Paul Miller is self-taught in electronics and many dialects of BASIC and is learning assembly language. He started on a TRS-80 Model III and has been programming for four years. He is currently writing a text Adventure game that he plans to market upon completion. Paul can be contacted at Route 1, Box 341, Wirtz, VA 24184, (703) 721-2910. (Please enclose an SASE when writing for a reply.)



Sanyo Synthesizer

By Paul T. Miller

and produces a "sharpening" or "flattening" of the note. Pressing the down arrow restores the note to its original tone. However, when tuning, you also change the pitch of all of the notes played. When a tune is played or saved, the TUNE value is applied, so if you tune the notes then save a composition, you won't have to retune it before playing the composition again.

Is your beat a little off at times? The best thing to do is use a metronome, a little box that makes a rythmic clicking sound. There is a "metronome" built in to *Sanyo Synthesizer*! Pressing the asterisk (*) at any time in the PLAY or RECORD modes toggles it on and off. Pressing the upper-left arrow (home key) and the up arrow slows down or speeds up the beat, thus the audible background beep you hear when the metronome is on.

To have more features, I added several function key redefinitions in Line 170. Pressing PF1 through PF5 produces short little scales and cute fanfares. This is accomplished by using the KEY n,string$ statement. The function keys are set to letters and numbers that correspond to those on the keyboard. To make your own, use up to eight letters or numbers in the KEY string. (It's a good idea to make the last character of the string a space so the last note won't keep playing.)

Programming Techniques

The CONT N at the beginning of the program disables the BREAK key, so if the BREAK key is accidentally pressed, you won't erase a tune from memory.

The KEY SAVE"KEY.KEY" at Line 170 saves the current function key values to the filename KEY.KEY on disk before it redefines the function keys. When you quit the program, the KEY LOAD"KEY.KEY" loads the original values back into BASIC.

I originally wrote the program so that after pressing a key it would loop through a list of note and key values until it found the right one and played it. This got very slow when pressing the higher valued keys. I replaced this with a routine that goes straight to the correct value — much like a random file, while the other version ran like a sequential file. This is accomplished by making a DATA table (Line 200) with the correct note value in the same position relative to the key that performs that value. When you press a key, its ASCII value is the position in the DATA table in which the value for that note is kept.

The metronome beep is accomplished by the OUT in Line 230. This port produces simple clicks and only three pure tones. I used the highest pure tone, but if you would like the beep lower, a value of 19 will work. A value of '4' is the lowest tone.

Sanyo Synthesizer was originally written for use with 256K of memory. This provides much longer tunes. If you have 256K, then change the CLEAR 20,,&H800 in Line 180 to CLEAR 20,,&H1F00. This reserves *much* more memory for arrays. Now change the DIM P(93),S(15000),CO(1000) in Line 180 to DIM P(93),S(30000),CO(5000). This provides compiled tunes with up to 5,000 notes and rests.

Secondary Features

When you want to play, compile, save or erase a tune from memory, the program checks to see if one has been recorded first. If not, the function is not even selected.

When saving a tune, if a tune with the same filename is already on the disk, the program asks if you want to overwrite it or not.

When loading a tune, if the tune is not on the disk, the function is automatically aborted.

If you select 'N' for new or 'Q' for quit, the program checks to see if that's what you really want by asking Are you sure? (Y/N) and waiting for a response.

Anytime the program is playing back a tune, you may abort it by pressing ESCape.

Implemented Tunes

If you would like to use these tunes in a BASIC program, it is relatively simple. First, POKE the machine language sound driver data at Line 190 into memory locations 100H through 13BH, then load the tune into an array. Loop through the array and play the tune. Listing 2 contains all you need to implement tunes into a BASIC program.

If you don't like the pulsing sound of the tunes, a compiled version of your tune can be loaded into a BASIC program. Listing 3 compiles a tune then resaves it to disk. Listing 4 is an example of loading a compiled tune for use in your own program.

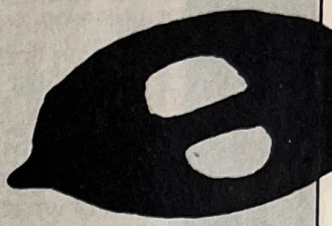
You can place the tune loader and playback lines in a GOSUB routine and define the filename before calling the routine. It would then be completely self-contained and work for any tune.

For those who have 256K, I have written another version of *Sanyo Synthesizer* that loads a high resolution graphics picture of the keyboard and menu. This picture has all of the keys labeled, with naturals white, sharps black and unused keys blue. If you would also like a copy of this version and you don't get SOFT SECTOR ON DISK, I will send both versions plus a couple of tunes I have saved myself. Just send a formatted disk and $5 for duplication time and shipping and handling to the address in the program. If you have any problems, want to modify the program or have already done so, I would certainly like to know. Feel free to call me anytime during the week after 4 p.m.

Listing 1:

```
1Ø '*** "SANYO SYNTHESIZER" - filename "SANYOSYN.BAS"
2Ø '    By Paul T. Miller - (7Ø3) 721-291Ø - Rt.1 Box 341, Wirtz, Va. 24184
3Ø '    copyright (c) 1985 by "The Sanyo Station"
4Ø CONT N:CLS:LOCATE 15,12,Ø
5Ø SYMBOL(11Ø,1),"SANYO SYNTHESIZER",3,2,1:SYMBOL(112,2),"SANYO SYNTHESIZER",3,2
,3:SYMBOL(114,3),"SANYO SYNTHESIZER",3,2,4:SYMBOL(25Ø,2Ø),"by Paul T. Miller",1,
1,6
6Ø FOR I=75 TO 54Ø STEP 3Ø:LINE(I,3Ø)-(I+25,4Ø),7,BF:NEXT:LINE(75,45)-(115,55),7
,BF:FOR I=12Ø TO 42Ø STEP 3Ø:LINE(I,45)-(I+25,55),7,BF:NEXT:LINE(I,45)-(I+4Ø,55)
,7,BF:LINE(75,6Ø)-(125,7Ø),7,BF:FOR I=13Ø TO 455 STEP 3Ø:LINE(I,6Ø)-(I+25,7Ø),7,
BF:NEXT
7Ø LINE(I,6Ø)-(I+4Ø,7Ø),7,BF:LINE(I+35,45)-(I+45,55),7,BF:LINE(I+45,45)-(I+9Ø,7Ø
),7,BF:LINE(75,75)-(11Ø,85),7,BF:FOR I=115 TO 455 STEP 3Ø:LINE(I,75)-(I+25,85),7
,BF:NEXT:LINE(I,75)-(I+45,85),7,BF:LINE(I+5Ø,75)-(I+75,85),7,BF
8Ø LINE(13Ø,9Ø)-(455,1ØØ),7,BF:LINE(17Ø,9Ø)-(173,1ØØ),Ø,BF:LINE(412,9Ø)-(415,1ØØ
),Ø,BF
9Ø FOR I=1 TO 8:READ X,I$:SYMBOL(X,32),I$,1,1,Ø:NEXT:DATA 14Ø,F#,17Ø,G#,2ØØ,A#,2
6Ø,C#,29Ø,D#,35Ø,F#,38Ø,G#,41Ø,A#:FOR I=13Ø TO 43Ø STEP 3Ø:READ I$:SYMBOL(I,47),
I$,1,1,Ø:NEXT:DATA F,G,A,B,C,D,E,F,G,A,B:SYMBOL(465,47),"C",1,1,Ø
1ØØ FOR I=1 TO 7:READ X,I$:SYMBOL(X,62),I$,1,1,Ø:NEXT:DATA 165,C#,195,D#,255,F#,
285,G#,315,A#,375,C#,4Ø5,D#:FOR I=155 TO 43Ø STEP 3Ø:READ I$:SYMBOL(I,77),I$,1,1
,Ø:NEXT:DATA C,D,E,F,G,A,B,C,D,E:SYMBOL(465,47),"C",1,1,Ø
11Ø SYMBOL(5Ø2,47),"RECORD",1,1,Ø:SYMBOL(512,62),"PLAY",1,1,Ø:SYMBOL(255,92),"ST
OP NOTE",1,1,4
12Ø COLOR 6:PRINT "MENU:";:COLOR 3:PRINT " selected by pressing ";:COLOR 7:PRINT
 "ESC":PRINT:FOR I=1 TO 7:READ I$:PRINT TAB(13);CHR$(34);:COLOR 6:PRINT I$;:COLO
R 7:PRINT CHR$(34):NEXT:DATA P,C,S,L,N,Q,ESC
13Ø COLOR 3:LOCATE 17,1:FOR I=1 TO 6:READ I$:PRINT TAB(18);"- ";I$:NEXT:PRINT TA
B(18)"- return to RECORD/PLAY mode":DATA play back tune,compile & play tune,save
 tune to disk,load tune from disk,new - erase tune from memory,quit - end progra
m
14Ø LOCATE 15,48:COLOR 6:PRINT "TUNE:":FOR I=1 TO 3:READ C,I$:COLOR 7:PRINT TAB(
49);CHR$(34);:COLOR 6:PRINT CHR$(C);:COLOR 7:PRINT CHR$(34);:COLOR 3:PRINT " - "
;I$:NEXT:DATA 26,make sharper,27,make flatter,25,restore
15Ø LOCATE 2Ø,49:COLOR 6:PRINT "METRONOME:":FOR I=1 TO 2:READ C,I$:COLOR 7:PRINT
 TAB(52);CHR$(34);:COLOR 6:PRINT CHR$(C);:COLOR 7:PRINT CHR$(34);:COLOR 3:PRINT
" - ";I$:NEXT:DATA 42,start/stop,24,speed up
16Ø COLOR 7:PRINT TAB(49);CHR$(34);:COLOR 6:PRINT "HOME";:COLOR 7:PRINT CHR$(34)
;:COLOR 3:PRINT " - slow down";
17Ø KEY SAVE"KEY.KEY":KEY 1,"ZXCVBNM ":KEY 2,",./QWER ":KEY 3,"][POIUY ":KEY 4,"
Z,T]T,Z ":KEY 5,"BB BVCX "
18Ø OPTION BASE 1:CLEAR 2Ø,,&H8ØØ:DEFINT A-Z:DIM P(1ØØ),S(15ØØØ),CO(1ØØØ):B=1:C=
Ø:MM=15:DEF SEG=SEG(4):FOR A=&H1ØØ TO &H13B:READ I$:POKE A,VAL("&H"+I$):NEXT:FOR
 I=44 TO 93:READ P(I):NEXT
19Ø DATA EB,2,Ø,Ø,FA,8B,EC,1E,C5,5E,8,8B,F,C5,5E,4,8B,17,E,1F,2E,89,E,2,1,B8,35,
Ø,34,8,E6,3A,FE,CC,75,3,4A,74,9,E2,F7,2E,8B,E,2,1,EB,EC,34,8,3C,35,75,2,E6,3A,1F
,CA,8,Ø
2ØØ DATA 311,86,277,248,97,Ø,22Ø,196,175,Ø,147,13Ø,Ø,1Ø9,Ø,262,Ø,Ø,Ø,Ø,Ø,Ø,417,4
99,528,185,Ø,44Ø,391,116,351,Ø,294,331,371,1Ø4,92,233,165,595,155,123,469,2Ø8,56
1,138,622,82,Ø,77
21Ø LOCATE 24,1:COLOR 6:PRINT "mode:":COLOR 7
22Ø D=15:N=Ø:R=Ø:LOCATE 25,1:PRINT "PLAY      ";
23Ø IF M=1 THEN MC=MC+1:IF MC>=MM-MN THEN OUT 56,2Ø1:MC=Ø
24Ø A$=INKEY$:MN=Ø:IF A$=""AND N=1 THEN E=T+C:MN=7:CALL &H1ØØ,E,D:GOTO 33Ø
25Ø IF A$=""THEN FOR I=1 TO 15:NEXT:T=Ø:GOTO 33Ø
26Ø A=ASC(A$):IF A=32 THEN N=Ø:GOTO 23Ø ELSE IF A=27 THEN 35Ø ELSE IF A=13 THEN
N=Ø:R=1-R:LOCATE 25,1:IF R=Ø THEN PRINT "PLAY     ";:GOTO 23Ø ELSE IF R=1 THEN P
RINT "RECORD   ";:GOTO 23Ø
27Ø IF A=42 THEN M=1-M
28Ø IF A=28 THEN C=C+1 ELSE IF A=29 THEN C=C-1 ELSE IF A=31 THEN C=Ø
29Ø IF A=12 THEN MM=MM+1 ELSE IF A=3Ø THEN MM=MM-1:IF MM<Ø THEN MM=Ø
3ØØ IF A>97 THEN A=A AND 223
31Ø IF A>93 THEN 23Ø ELSE IF P(A)=Ø THEN 23Ø ELSE IF P(A)=T THEN S(B)=Ø:B=B+1
32Ø T=P(A):E=T+C:CALL &H1ØØ,E,D:N=1
33Ø IF R=1 THEN S(B)=T:B=B+1
34Ø GOTO 23Ø
35Ø D=15:N=Ø:R=Ø:GOSUB 57Ø:PRINT "SELECT:   ";:LOCATE 25,1,Ø:A$=INPUT$(1):IF ASC
(A$)>96 THEN A$=CHR$(ASC(A$)-32)
36Ø IF A$=CHR$(27)THEN GOSUB 57Ø:GOTO 22Ø
37Ø IF A$="Q"THEN PRINT "Quit...are you sure? (Y/N)";:GOTO 5ØØ
38Ø IF A$="C"AND B>1 THEN PRINT "Compiling...";:I=Ø:REC=Ø:NO=1:GOTO 58Ø
39Ø IF A$="P"AND B>1 THEN PRINT "Playing Back...ESC to stop";:GOTO 44Ø
4ØØ IF A$="N"AND B>1 THEN PRINT "New... are you sure? (Y/N)";:GOTO 46Ø
41Ø IF A$="S"AND B>1 THEN PRINT "Save...enter filename: ";:GOTO 47Ø
```

```
42Ø IF A$="L"THEN PRINT "Load...enter filename: ";:GOTO 49Ø
43Ø BEEP:GOTO 35Ø
44Ø F=D+2:FOR I=1 TO B-1:T=S(I):IF T=Ø THEN FOR Z=1 TO 15:NEXT ELSE E=T+C:CALL &
H1ØØ,E,F
45Ø IF INKEY$=CHR$(27)THEN BEEP:GOTO 35Ø ELSE NEXT:GOTO 35Ø
46Ø A$=INPUT$(1):IF A$="Y"OR A$="y"THEN GOSUB 57Ø:B=1:GOTO 22Ø ELSE GOTO 35Ø
47Ø GOSUB 51Ø:GOSUB 57Ø:OPEN"R",1,F$+".SYN":IF LOF(1)>Ø THEN CLOSE:PRINT "File e
xsists. Overwrite? (Y/N)";:A$=INPUT$(1):IF A$="Y"OR A$="y"THEN 48Ø ELSE 35Ø
48Ø CLOSE:OPEN"O",1,F$+".SYN":GOSUB 57Ø:PRINT "Saving...";:PRINT #1,C:PRINT #1,B
-1:FOR I=1 TO B-1:PRINT #1,S(I):NEXT:CLOSE:GOTO 35Ø
49Ø GOSUB 51Ø:OPEN"R",1,F$+".SYN":IF LOF(1)=Ø THEN CLOSE:KILL F$+".SYN":GOTO 35Ø
:ELSE CLOSE:GOSUB 57Ø:PRINT "Loading...";:OPEN"I",1,F$+".SYN":INPUT #1,C:INPUT #
1,B:FOR I=1 TO B:INPUT #1,S(I):NEXT:CLOSE:GOTO 35Ø
5ØØ A$=INPUT$(1):IF A$="Y"OR A$="y"THEN CLS:KEY LOAD"KEY.KEY":KILL"KEY.KEY":COLO
R 2,Ø:STOP ELSE GOTO 35Ø
51Ø F$="":LOCATE 25,24,1
52Ø A$=INPUT$(1):A=ASC(A$):IF A=8 AND LEN(F$)=Ø THEN 52Ø ELSE IF A=8 THEN PRINT
CHR$(8)" "CHR$(8);:F$=LEFT$(F$,LEN(F$)-1):GOTO 52Ø
53Ø IF A=13 AND LEN(F$)=Ø THEN GOTO 35Ø ELSE IF A=13 THEN F$=LEFT$(F$,1Ø):LOCATE
 25,1,Ø:RETURN
54Ø IF A>97 THEN A=A AND 223
55Ø IF A=32 OR A=46 THEN 52Ø
56Ø IF LEN(F$)=1Ø THEN BEEP:GOTO 52Ø ELSE PRINT A$;:F$=F$+A$:GOTO 52Ø
57Ø LOCATE 25,1:PRINT STRING$(4Ø," ");:LOCATE 25,1:RETURN
58Ø I=I+1:IF I=B THEN 61Ø ELSE REC=REC+1:CO(REC)=S(I)
59Ø IF S(I)=S(I+1)THEN NO=NO+1:I=I+1:IF I=B THEN 61Ø ELSE 59Ø
6ØØ REC=REC+1:CO(REC)=NO*16:NO=1:GOTO 58Ø
61Ø LOCATE 25,1:PRINT "Playing compiled...ESC to stop";:FOR I=1 TO REC-1 STEP 2:
T=CO(I):D=CO(I+1):IF T=Ø THEN FOR Z=1 TO D:NEXT ELSE E=T+C:CALL &H1ØØ,E,D
62Ø IF INKEY$=CHR$(27) THEN BEEP:GOTO 35Ø ELSE NEXT:GOTO 35Ø
```



Listing 2:

```
5 'LISTING4.BAS - loads compiled tune for use in BASIC program
9 'set up sound driver and variables
1Ø CLEAR 2Ø:DEFINT A-Z:DEF SEG=SEG(4):FOR A=&H1ØØ TO &H13B:READ I$:POKE A,VAL("&
H"+I$):NEXT:DIM CO(1ØØØ):F$="TUNE"
2Ø DATA EB,2,Ø,Ø,FA,8B,EC,1E,C5,5E,8,8B,F,C5,5E,4,8B,17,E,1F,2E,89,E,2,1,B8,35,Ø
,34,8,E6,3A,FE,CC,75,3,4A,74,9,E2,F7,2E,8B,E,2,1,EB,EC,34,8,3C,35,75,2,E6,3A,1F,
CA,8,Ø
29 'load it
3Ø OPEN"I",1,F$+".CMP":INPUT #1,C:INPUT #1,R:FOR I=1 TO R:INPUT #1,CO(I):NEXT:CL
OSE
39 'play it
4Ø FOR I=1 TO R STEP 2:T=CO(I):D=CO(I+1)+5:IF T=Ø THEN FOR Z=1 TO D:NEXT ELSE E=
T+C:CALL &H1ØØ,E,D
5Ø NEXT:END
```

Listing 3:

```
5 'COMPILER.BAS - compiles "pulse" tunes into pure tunes
1Ø DEFINT A-Z:CLEAR Ø,,&H8ØØ:DIM S(15ØØØ),CO(1ØØØ):R=Ø:NO=1:CLS:PRINT "Sanyo Syn
thesizer files on default drive are:":PRINT:FILES"*.SYN":PRINT:INPUT"Name of tun
e to be compiled";F1$
2Ø PRINT:INPUT"Name of output filename (RETURN for same name)";F2$:IF F2$=""THEN
 F2$=F1$
3Ø PRINT "Loading...":OPEN"I",1,F1$+".SYN":INPUT #1,C:INPUT #1,B:FOR I=1 TO B:IN
PUT #1,S(I):NEXT:CLOSE:PRINT "Compiling...":I=Ø
4Ø I=I+1:IF I=B THEN 7Ø ELSE R=R+1:CO(R)=S(I)
5Ø IF S(I)=S(I+1)THEN NO=NO+1:I=I+1:IF I=B THEN 7Ø ELSE 5Ø
6Ø R=R+1:CO(R)=NO*16:NO=1:GOTO 4Ø
7Ø PRINT "Saving...":OPEN"O",1,F2$+".CMP":PRINT #1,C:PRINT #1,R-1:FOR I=1 TO R-1
:PRINT #1,CO(I):NEXT:CLOSE
8Ø BEEP:PRINT "Done!":END
```

Listing 4:

```
5 'LISTING2.BAS - loads tune for use in BASIC program
9 'set up sound driver and variables
1Ø CLEAR 2Ø:DEFINT A-Z:DEF SEG=SEG(4):FOR A=&H1ØØ TO &H13B:READ I$:POKE A,VAL("&
H"+I$):NEXT:DIM S(15ØØØ):D=17:F$="TUNE"
2Ø DATA EB,2,Ø,Ø,FA,8B,EC,1E,C5,5E,8,8B,F,C5,5E,4,8B,17,E,1F,2E,89,E,2,1,B8,35,Ø
,34,8,E6,3A,FE,CC,75,3,4A,74,9,E2,F7,2E,8B,E,2,1,EB,EC,34,8,3C,35,75,2,E6,3A,1F,
CA,8,Ø
29 'load it
3Ø OPEN"I",1,F$+".SYN":INPUT #1,C:INPUT #1,B:FOR I=1 TO B:INPUT #1,S(I):NEXT:CLO
SE
39 'play it
4Ø FOR I=1 TO B:T=S(I):IF T=Ø THEN FOR Z=1 TO 15:NEXT ELSE E=T+C:CALL &H1ØØ,E,D
5Ø NEXT:END
```

soft sector

Here we are again with one more winner of The Great SOFT SECTOR One-Liner Contest.

To briefly restate the rules of the contest, begin the program with a line number and end it in a single line. Anything else goes. Entries will be accepted in either Sanyo BASIC or GW-BASIC. Please include a printed listing, a title for the program and a short explanation of what it does. Send it to The Great SOFT SECTOR One-Liner Contest, P.O. Box 385, Prospect, KY 40059.

one-liner contest

One Liner

Color Combinations

This program demonstrates all of the BASIC COLOR combinations from which one might choose for special effects. One less character or space in the PRINT statement will cause the combinations to be single-spaced. If you lose count, a few extra spaces will insure double-line spacing. If you have a monochrome monitor, you will be surprised to get four colors and many combinations.

```
1 COLOR X,Y:PRINT "       COLOR "X","Y"
     By Richard A. Milroy of NAMFONOS IN
CORPORATED          ":FOR T=1 TO 25ØØ:NEX
T:X=X+1:IF X<8 THEN GOTO 1 ELSE X=Ø:Y=Y+
1:IF Y>7 THEN Y=Ø: GOTO 1 ELSE COLOR Y,Y
:PRINT :PRINT :FOR T=1 TO 25ØØ:NEXT:GOTO
 1
```

Dick Milroy
Annapolis, MD

(For this winning one-liner contest entry, the author has been sent a copy of the June edition of SOFT SECTOR ON DISK).

Can you still afford that chauffeur with your pink Cadillac? This program answers these questions

Drive it, Park it, Buy it or Sell it?

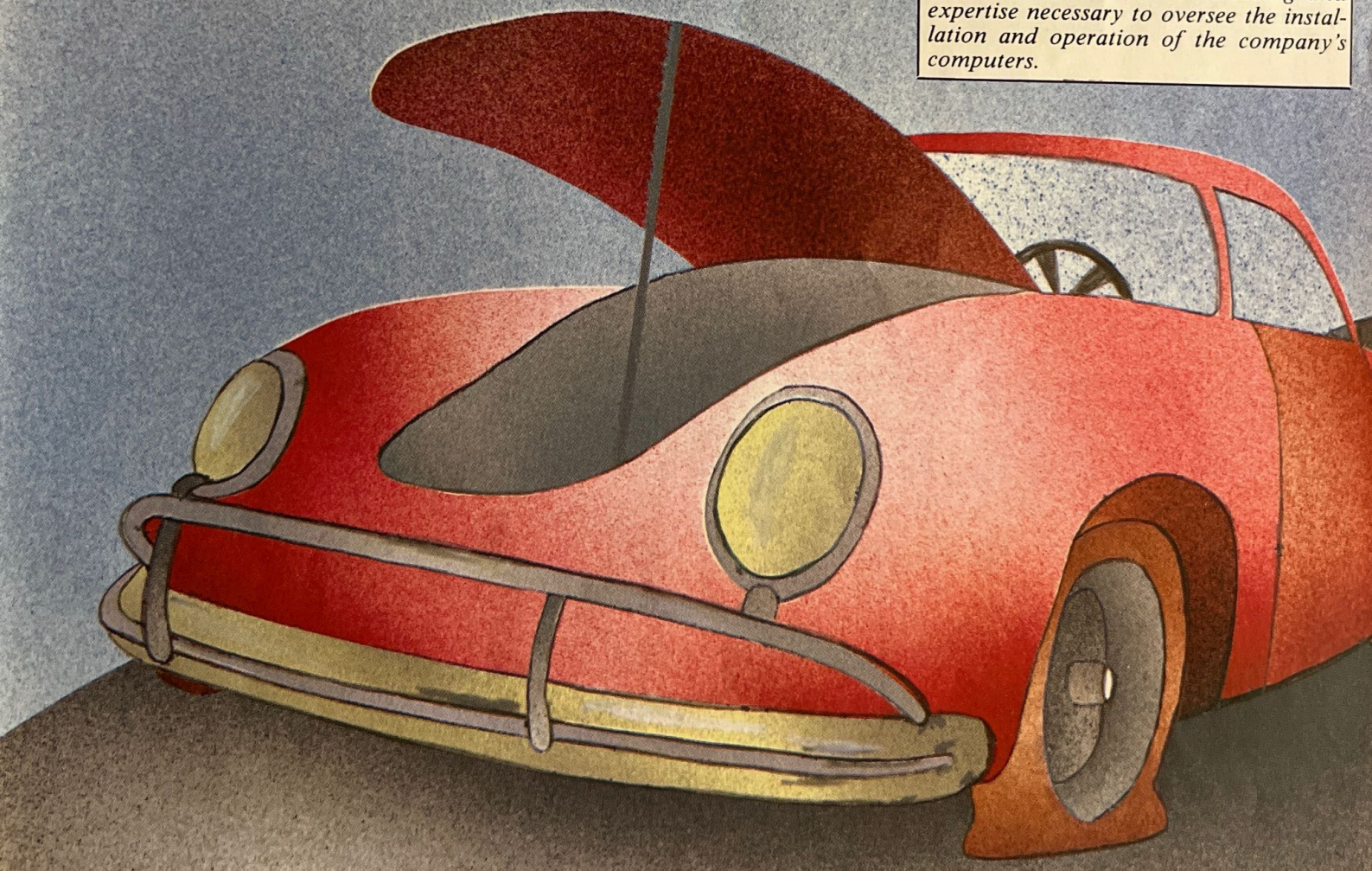
By A. Richard Baines

CarCost helps answer some of those hard-to-answer questions: "What does it cost to operate my car? Should I keep this one, or would a new car be more economical? If I do buy a new car, will I be able to support my computer habit in the style to which I have become accustomed?"

This simple, self-instructing program gives the user a good indicator for answering these questions. *CarCost* is not intended to give an actual out-of-pocket accounting for every vehicle, but it will give the average cost of operating that vehicle.

I hope other readers will be able to expand from this program to produce other programs that compare facts and figures.

Richard Baines, a confessed car nut, is a civil engineering designer for a company near Baltimore, Maryland. His computer hobby has provided the training and expertise necessary to oversee the installation and operation of the company's computers.



The listing:

```
10 CLS
20 PRINT:PRINT:PRINT" ARB'S  VEHICLE OPERATING COST"
30 PRINT:PRINT"THIS PROGRAM WILL TELL YOU THE COST OF OPERATING YOUR VEHICLE"
40 PRINT"BASED ON:"
50 PRINT SPC(9)"GAS MILEAGE"
60 PRINT SPC(9)"EXPECTED REPAIRS"
70 PRINT SPC(9)"INSURANCE COSTS"
80 PRINT SPC(9)"EXPECTED VEHICLE LIFE"
90 PRINT:PRINT:PRINT"ANSWER THE FOLLOWING QUESTIONS FOR YOUR VEHICLE"
100 PRINT:PRINT"FOR DOLLAR ENTRIES, DO NOT USE DOLLAR SIGN OR COMMA."
110 PRINT " "
120 INPUT "VEHICLE NAME";V$
130 INPUT "EXPECTED VEHICLE LIFE (100000 MILES?)";L
140 PRINT "VEHICLE COST (PURCHASE PRICE MINUS THE VALUE YOU EXPECT THE VEHICLE T
O HAVE AT":PRINT L" MILES)":INPUT VC
150 INPUT "MILEAGE YOU EXPECT TO PUT ON THE VEHICLE IN THE NEXT YEAR";M
160 INPUT "VEHICLE MILEAGE WHEN PURCHASED ";VM
170 INPUT "COST OF THE GASOLINE YOU PLAN TO USE IN THE VEHICLE";G
180 INPUT "VEHICLE M.P.G.";MPG
190 INPUT "EXPECTED YEARLY REPAIR BILL ";R
200 INPUT "YEARLY INSURANCE COST";I
205 LPRINT:LPRINT CHR$(27)+CHR$(14);TAB(18);"CARCOST":LPRINT
210 LPRINT "THIS COST IS NOT THE ACTUAL OUT OF POCKET EXPENSES, BUT AN ESTIMATE
OF OVERALL  COSTS OVER THE VEHICLE LIFE"
220 LPRINT:LPRINT;CHR$(27)+CHR$(14);V$
230 LPRINT:LPRINT "OVERALL VEHICLE COST $";VC
240 LPRINT "EXPECTED VHICLE LIFE";L
250 LPRINT "EXPECTED YEARLY MILEAGE ";M
260 LPRINT "VEHICLE MILEAGE WHEN PURCHASED ";VM
270 LPRINT "GASOLINE COST PER GALLON $";G
280 LPRINT "VEHICLE M.P.G. ";MPG
290 LPRINT "EXPECTED YEARLY REPAIR BILL $";R
300 LPRINT "YEARLY INSURANCE COST $";I
310 LPRINT:LPRINT"THE AVERAGE COST TO OPERATE YOUR ";V$
320 LPRINT"WILL BE:"
330 TC=((VC/(L-VM)+(G/MPG)+(I/M)+(R/M)))
340 LPRINT USING "$#.##";TC
350 LPRINT "PER MILE"
360 LPRINT:LPRINT"THE AVERAGE YEARLY COST TO OPERATE YOUR ";V$
370 LPRINT "WILL BE:"
380 YC=M*TC
390 LPRINT USING "$####.##";YC
400 LPRINT "PER YEAR"
410 LPRINT:LPRINT:LPRINT:LPRINT:LPRINT:LPRINT:LPRINT
420 GOTO 10
430 END
```

DELPHI BUREAU

Last month we began our investigation of the personal Workspace area in the MS-DOS SIG on Delphi. Primarily, we discussed how to upload a file into the Workspace using one of the three supported transfer protocols: ASCII, Xmodem and Kermit. Now let's take a look at what you can do with a file after it's there.

One of the most important things, of course, is to publish a program in the database so that all the other members can download it. This can be one of the generally excellent "user-supported" or "Shareware" programs, or it can be a program that you've written yourself and want to share with others. After all, it's hard not to be proud of a program you've created, slaving over the keyboard, running it over and over until it's just right. And it's a lot more fun when you can share your work with others who will appreciate it as much as you do.

After you have entered your Workspace, the first thing you may want to do prior to making a database submission is to check the directory to ascertain the proper filename of the programs you're about to submit. To do this, you type DIR, just as on your own computer. And remember, too, that the MS-DOS question mark (?) and asterisk (*) wild cards work the same familiar way when used with the DIR command in your Workspace.

After you've checked for the proper filenames, you're ready to begin the submission process. Type SUBMIT. The system responds with a couple of lines of instruction and asks if you wish to continue. Of course, you respond YES, or with a simple Y to save yourself a few keystrokes.

Now the system asks you how many files you will be submitting. One of the nice things about the Delphi database software is that it allows you to "group" related files together under a single description. Let's say you have three files that you're submitting together: a compiled, executable version of your BASIC program; the BASIC source code itself; and a separate file of instructions and documentation. That's three, so you respond with 3 when the system requests this information.

The system then asks you whether all three files are related in such a way that they should be listed as a single "group" in the directory. In this case, you again respond with YES.

The system then asks you for a "filetype," and displays a list from which to choose: program or program pack, newsletter, article, transcript, documentation, data (graphics, etc.) or miscellaneous text. Since we're dealing with a program here, the proper response is PROGRAM, or simply PRO.

Now the system asks what topic of the database you wish to submit your file to. If you're not familiar with the topics that are available, you can enter a question mark (?) here and the topics will be displayed:

General Information	Archives
Business	Education
Home & Games	Programming
Telecommunications	Utilities
PCM Collection	Info on PCM
Soft Sector Info	SS On Disk

If your submission happens to be a recipe file program, for example,we enter HOM to select the Home & Games topic.

Now a name for the "group/set" is requested. This is the name that appears in the database directory, but you're not limited to a directory entry as you may be accustomed. Here you're allowed up to 32 characters, so you can really use a meaningful name, such as RECIPE FILE DATABASE.

When you're finished naming the group, you are asked to enter a brief description of the file. This is where you should write a little paragraph telling all about your program: the author's name, what the program does, the specific system requirements; anything you can think of that the person who uses it may need to know. When you're finished, enter a CTRL-Z to close the description entry.

Now comes the time when you must enter "keywords" that describe your program. These are simply descriptive words that can later be searched on to locate programs of similar type. The entire directory in a certain topic can be extremely long and confusing, but by searching on these keywords, you can set it to display only those files that are similar to something you're looking for.

The first keyword requested is the "primary" keyword, and must be chosen from a select list that has been installed by the database manager. At the primary keyword prompt, you can once again enter a question mark (?) to display the choices. In the case of the Home & Games topic, the choices are: arcade games, adventure games, finance, management and miscellaneous. In this case, it looks like miscellaneous might be the best choice for our recipe file program, so enter MISC.

You can now enter any other keywords that you like, the idea being to try to imagine what keyword someone else looking for a recipe file program might try to search for. Some possibilities would be "cooking," "recipes," "database," or maybe even "food." You can also put the author's name here and something like "VB" if the program requires the Video RAM Board to run on the Sanyo 555.

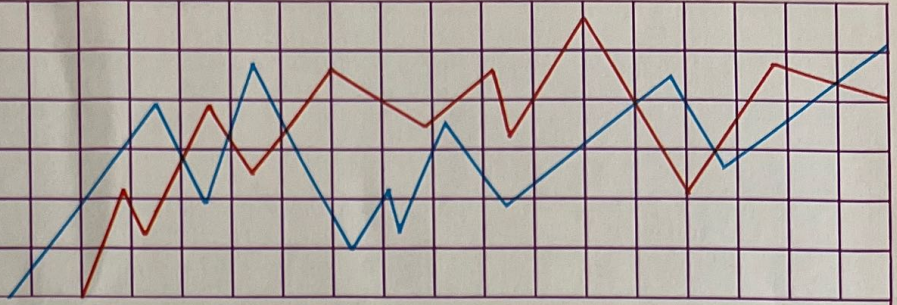
Next you are asked for information about each of the files. The first request is for the filename of the file in your Workspace. You found this earlier when you checked it in the directory. Then you are asked if the file must have any special filename when it is downloaded — generally the same as you uploaded it into your Workspace, but not necessarily. Finally, you are asked for a name to appear with the file in the group directory listing which, again, can be a descriptive name, such as RECIPE FILE DOCUMENTATION.

This naming process is repeated for each of the three files that you are submitting, along with a query as to whether you would like it deleted from your Workspace. When you finish with the filename information for each file, the submission process is complete. Your file then goes to a "preview" area for testing before being moved into the open database for the members.

Although going through all this may seem like a very long, drawn-out process, it really isn't. After you've tried it once, you'll see just how simple it really is.

— Kevin Nickols
MS-DOS SIG Manager

BUSINESS Sector



CHARLOTTE & BRIAN STONE
Soft Sector Contributing Editors

Q. *Is there any way to combine two or more* WordStar *files into a single new file? Also, is it possible to transfer selected blocks of text from one file to another?*

Tom Hettinger
Berlin, NH

A. There is an easy way to merge two files with *WordStar* into a new file.

1) Load the file you would like to be the beginning of the new file.

2) Go to the end of the file just loaded or to a mid-point in the file where you want the second file to be inserted.

3) Do CONTROL-KR (^KR). You are then asked NAME OF FILE TO READ; enter the name of the second file (the one you want to merge), then press RETURN. You can add a drive name such as B:FILENAME if necessary.

Transferring selected blocks of text from one file to another is just as easy as merging.

1) Load the file that has the block of text you would like to use.

2) Mark the beginning of the block with CONTROL-KB (^KB); find the end of the block and mark it with CONTROL-KK (^KK). This puts the marked text in inverse video.

3) Next do CONTROL-KW (^KW) to write text. You are asked NAME OF FILE TO WRITE. Insert your *new* filename with a drive specifier if necessary and press RETURN. This file can then be read into any other file as necessary.

4) This method can be used to move text within a file also by using the CONTROL-KV (^KV) "move" or CONTROL-KC (^KC) "copy" command.

The best way to learn these functions is to try them on copies of existing files. Back up a data disk and experiment.

Q. *One of the drawbacks of* WordStar *for my work is the limited ability to send control codes to the printer. The control codes I use frequently are as follows: subscript, superscript, italics, double spacing, emphasized print, elite type, near-letter quality (on the Panasonic 1091) and paper end detect. Is there any way to enter all these codes through* WordStar?

Erwin M. Reimann, Ph.D.
Toledo, OH

A. The functions you require can be made available by using the *WordStar Install* program. Subscript, superscript, italics, double space, emphasized print, elite print, and near-letter quality print can function for your Panasonic KX-P1091 printer. First install *WordStar* as an Epson printer from the install menu, then continue to the custom install menu and make the following changes as necessary. All of these functions can be changed to whatever you like — *WordStar* does not care what you are sending to the printer from a given control sequence.

Desired Function	Patch Location
Insert 1091 Code	SuperScript ^PV
Insert 1091 Code	SubScript ^PT
Insert Italic Code	Ribbon Select ^PY for on/off
Insert Emphasized ON	Phantom Space ^PF
Insert Emphasized OFF	Phantom Rubout ^PG
Insert Elite Code ON/OFF	Character Pitch ^PA and ^PN
Insert N.L.Q. Print Code	User Definable ^PQ and ^PW
Insert Paper End Code	User Definable ^PE and ^PR

Line spacing should be set before the text is generated. If line spacing is set with ^OS after text generation, you will have to reformat all of the text and the only part that is reformed will be the paragraphs with wrap-around lines, not lines terminated with a carrige return.

Q. *I use* WordStar *exclusively and think it's great. However, I am also using a Juki 6100 printer. I set up the printer as for the Diablo. Everything works great until I want to print documents several pages long. I must either wait about two minutes between pages (individual feed), or the second (or maybe third or fourth) page doesn't print. Instead, I get a blinking light on the printer control panel indicating communications error. The page in question begins to print, but stays in the upper left corner, constantly overstriking. Even if I wait several minutes between pages, I still have the problem. I must shut the computer down, reboot the system and restart printing at the appropriate page. No real problem for one- or two-page documents, but I'm writing my doctoral thesis!*

Reverend John R. Sittema
Pella, IA

A. There have been other letters about problems with the Juki 6100 printer much the same as yours. The only way I can help is to suggest trying another printer. If the other printer functions correctly, you have found the problem. If it does not function correctly, the best solution is to send the computer back to Sanyo in New Jersey for repair. There have been some instances where the printer ports on the computer have been found to be the problem.

Q. *I have made several unsuccessful attempts to install* WordStar *with the reduced-size subscripts and superscripts the Star Gemini-10X printer is capable of producing. The change from normal print mode to super/subscript is accomplished using codes* 1Bh 53h 00h *and* 1Bh 53h 01h *respectively. These values were installed in 'P' (carriage roll). The cancel code,* 1Bh 54h, *installed in the vertical movement section of* Install *does not operate; the printer continues to produce half-height characters.*

Can the superscript, subscript and cancel commands be programmed onto the P.F. keys?

Richard L. Pierr
Christchurch, New Zealand

A. You seem to be using the correct codes to toggle the printer from super/subscript to standard print. If you cannot get the Star Gemini-10X printer to return from super/subscript with the codes shown in the manual, you could use the printer reset code, which is ESCAPE 40 Hex or 1B 40 Hex. This causes a reset the same as if you just turned on the printer.

The only problem is if you also have another function setup, such as italics, it will be lost.

There are a large number of user-definable functions in *WordStar* and I suggest you use them rather than the programmable function keys.

Q. *How can I install* WordStar *for a Comrex CR-1 Comwriter printer? I just bought a Chaucer Software proportional spacing program for* WordStar *and can't get it to run properly.*

James R. Leong
Los Angeles, CA

A. The Comrex CR-1 Comwriter printer emulates the Diablo 1610 printer. If you install *WordStar* for this printer you should not have any problems, and the proportional spacing software should work correctly.

Q. *I'm having a problem with* WordStar *on the MBC-555. I deal with files of large size produced by* DataStar. *The files are so large the disk fills up before the* *.BAK *is complete.*

Is there a patch to force WordStar *to make its automatic backup on the other (non-default) disk? If not, can* WordStar *be forced to* not make *a backup?*

Robert H. Geeslin, Ed.D.
Tulsa, OK

A. The easiest way to keep *WordStar* from making the *.BAK file is as follows:

Load the file in the normal way, do whatever editing must be done, then rather than saving the file or exiting by ^KD, do a complete block write of the file onto itself. This is done by marking the beginning of the text with ^KB, the end of the text with ^KK, then doing a ^KW to write a block of text to disk. You will be asked for the filename to write; use the same name as the file you loaded and you can write over it, not creating a backup.

The real flaw with this method is if the computer ever screws up you stand a chance of losing the entire text file. The safest thing to do is work with a backup copy until you know it is correct, then transfer the backup to its desired location.

Q. *I purchased a letter-quality printer made in Japan. The model is Admate DSY-120. Could you please inform me as to what American brand letter-quality printer this printer is compatible with?*

Submitting Material

Contributions to SOFT SECTOR are welcome from everyone. We like to run a variety of programs that are useful/helpful/fun for other Sanyo owners.

- **FORMAT:** Unless the program accompanying your submission is less than 10 lines, we must have the program itself on disk. We will print out the listing to our specifications. We simply cannot take the time to key in (and debug our typing errors) material that is longer. Editorial copy can also be included on disk, using any of the word processors currently available for the Sanyo 550, 555 or 775. However, please also include a double-spaced hard copy of your editorial material and hard copy of your program listing. Please do not send text in all capital letters; use upper- and lowercase. While it is a big help to us in typesetting to receive your article saved on disk using the ASCII option, it is not mandatory. But we must have, at the very least, a double-spaced hard copy of the article.

- **WHAT TO WRITE:** Anything with a practical application. If it interests you, it will probably interest a lot of others. However, we prefer articles with accompanying programs that can be entered and run. We can prepare finished tables, diagrams and schematics from your rough draft *if* you provide legible copy and full directions. We have a continuing need for short articles with short listings.

We do pay for submissions, based on a number of criteria. Those wishing remuneration should *so state* when making submissions.

For the benefit of those who wish more detailed information on making submissions, please send a self-addressed, stamped envelope (SASE) to: Submissions Editor, SOFT SECTOR, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit programs or articles currently submitted to another publication.

soft sector

Back Issue Availability

Back copies of many issues of SOFT SECTOR are still available.

All back issues sell for the single issue cover price. In addition, there is a $2 charge for the first issue plus 50 cents for each additional issue mailed in the U.S. When possible, issues are mailed UPS. The postage cost in Canada and Mexico is $3 for the first issue and $1 for each additional issue.

VISA, MasterCard and American Express accepted. Kentucky residents please add five percent state sales tax. In order to hold down costs, we do not bill and no C.O.D. orders are accepted.

We suggest you order back issues you want now while supplies last.

To order, just fill out the form on the next page and mail it with your payment to:

Soft Sector
The Falsoft Building
P.O. Box 385
Prospect, KY 40059

BACK ISSUE ORDER FORM

☐ Please send me the following back issues:

(Payment must accompany back issue orders.)

VOLUME 1

NO.	MON.	YR.		PRICE	
1	AUG.	'84	PREMIER ISSUE	$3.00	☐
6	JAN.	'85		$3.00	☐
7	FEB.	'85		$3.00	☐
8	MAR.	'85		$3.00	☐
9	APR.	'85		$3.00	☐
10	MAY	'85		$3.00	☐
11	JUNE	'85		$3.00	☐
12	JULY	'85		$3.00	☐

VOLUME 2

1	AUG.	'85	$3.00	☐
2	SEPT.	'85	$3.00	☐
3	OCT.	'85	$3.00	☐
4	NOV.	'85	$3.00	☐
5	DEC.	'85	$3.00	☐
6	JAN.	'86	$3.00	☐
7	FEB.	'86	$3.00	☐
8	MAR.	'86	$3.00	☐
9	APR.	'86	$3.00	☐
10	MAY.	'86	$3.00	☐
11	JUNE	'86	$3.00	☐

TOTAL _____

SHIPPING & HANDLING* _____

KY RESIDENTS ONLY ADD 5% _____

TOTAL AMOUNT ENCLOSED _____

To order by phone (*credit card orders only*), call (800) 847-0309, 8 a.m. - 5 p.m. EST. All other inquiries call (502) 228-4492.

* See reverse side.

Save money on back issues. See the binder ad on Page 6.

NAME _____ ADDRESS _____ CITY _____ STATE _____ ZIP _____

☐ PAYMENT ENCLOSED OR ☐ CHARGE TO: ☐ VISA ☐ MC ☐ AE

SIGNATURE _____ CARD # _____ EXPIRATION DATE _____ PHONE # _____

currently have the "Install" on WordStar *set to Qume.*

Michael J. Mohtiak
APO
New York, NY

A. I have never heard of an Admate DSY-120 printer. But since you have purchased it out of the country, this is not surprising.

It seems from all the references in the manual that the printer is Qume compatible. The best test is to install it and run the *WordStar* test file and see what happens. You say you have the software installed for the Qume — how does it work? If the output looks like you want it, it must be installed correctly.

Q. *I use a copy of* EasyWriter II *with an Okidata Microline 92 printer. If I had access to a Sanyo MBC 555-2 system hooked up with a daisy wheel, could I use a copy of* EasyWriter II *with the daisy wheel installation to run off a copy of a document originally created on* EasyWriter II *with Okidata installation?*

Recently, I decided to give WordStar *a test run to compare it with* EasyWriter II *and discovered it goes crazy when asked to print out a copy of a document. I guess my printer is incorrectly installed. I know very little about printer installation and am having real problems getting my system configured properly.*

Catherine K. Devitt
Montpelier, VT

A. It does not make any difference from what printer the document is printed. You can print the same letter on a daisy wheel or dot-matrix printer and the results should be the same as long as you do not use special codes for the daisy wheel printer that cannot be performed by the dot matrix printer.

If you install *WordStar* for the "standard printer" from the install menu, your letters will print. This is provided that the text has been saved in the ASCII format expected by *WordStar* and you do not use control codes *WordStar* cannot process.

Q. *In another magazine I read a letter saying the fastest way to print a* WordStar *file was to execute the following command structure:* `Ctrl-KD, P, Ctrl-R, Esc.` *I tried it and it worked. The only problem is that when I finish editing my next file and call for a print using this method, I end up printing my first file all over again instead of the one I have just edited. It seems my original file continues*

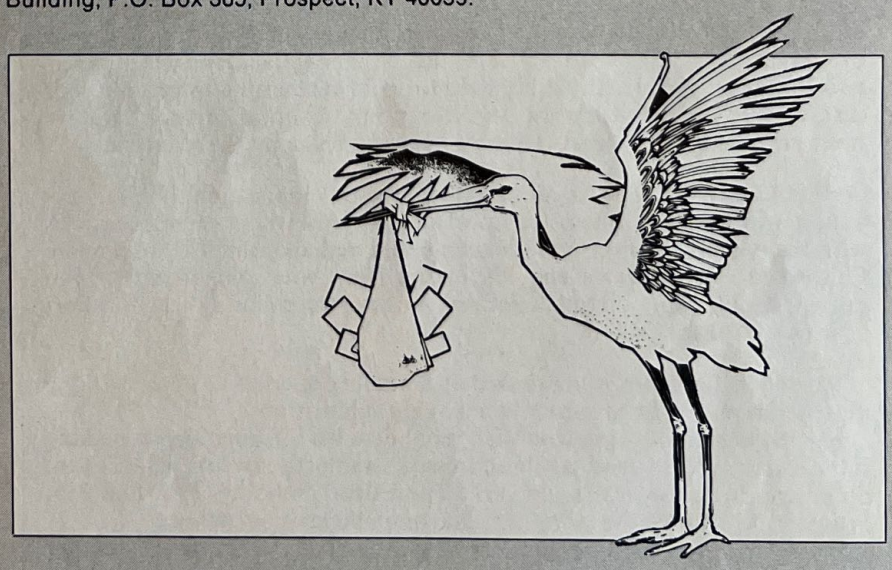
SUBSCRIPTION INFORMATION

Your copy of SOFT SECTOR is sent second class mail. If you do not receive your copy by the 15th* of the month of the cover date, send us a card and we will mail another. *Canadian subscribers and foreign air mail allow two additional weeks.

You must notify us of a new address when you move. Notification should reach us no later than the 10th of the month prior to the month in which you change your address. Sorry, we cannot be responsible for sending another copy when you fail to notify us.

Your mailing label also shows an "account number" and the subscription expiration date. Please indicate this account number when renewing or corresponding with us. It will help us help you better and faster.

For Canadian and other non-U.S. subscribers, there may be a mailing address shown that is different from our editorial office address. Do not send any correspondence to that mailing address. Send it to our editorial offices at The Falsoft Building, P.O. Box 385, Prospect, KY 40059.



to print unless I rename the file at the NAME OF FILE TO PRINT *prompt. Is there any way of getting around this seemingly unnecessary step in order to streamline the print function while in the edit mode?*

Robert C. Bliss
Lawrenceville, GA

A. You are correct; any use of ^KR returns the last name, not the current one or the one you expect. This function is a little spooky and unpredictable until it is used every day. When you get the hang of it you will only foul up once in a while. I have asked MicroPro about this feature and they (the actual program caretaker) say it works just the way it's intended. It does have its uses as I use it all of the time in my work.

Q. *When running* MailMerge — *particularly in the application of legal precedents — how can I protect an indent? As an example, when a variable is entered onto the left margin, the printed information is automatically wrapped to the left margin. When the variable* &name& *is entered, the result is printed on the left margin.*

Kay Nielsen
Hamilton, New Zealand

A. There is no way I know of to protect the indent as you require. The only option is to reduce the space of the indent so you can insert longer names.

Q. *In my 550/555 I have a* DataStar *file named* ANSUM *in Drive B: and the* DataStar *operating program in Drive A:. In the B: directory,* ANSUM *has BAK, DEF, DTA and NDX files shown. The problem is I can no longer invoke this file. With the* DataStar *operating disk in Drive A: and the destination disk, containing* ANSUM, *in Drive B:,* DataStar *is entered in A:. At the cursor prompt for filename,* B:ANSUM *is entered. The response is* ??? No key field has been established. Use ^K to assign key status. *Going through the similar invoking procedure for* FormGen *for this file, the response is "no such file exists." Is there any way I can call up this file form in* FormGen *in order to reestablish the key field?*

Rhett McMillian
New Smyrna Beach, FL

A. When the screen is displaying the response ??? No key field has been established. Use ^K to assign key status, you are in *FormGen. DataStar* has automatically transferred to *FormGen* because of a problem with your definition file. The rest of the prompt asks for an exit command and lists the *FormGen* exit alternatives. If you choose SPACE= continue without saving form, you will see the ANSUM form displayed on the screen and you will be able to check the status of the key field and reestablish it.

I don't understand a response that "no such file exists" when you try to invoke your ANSUM file with *FormGen*. If a file is named that has not yet been created when invoking *FormGen*, it opens a new file by whatever name you have typed and displays Help Screen #1. Then, if you type ^J, it displays an empty drawing screen. I would be interested to know exactly what you typed to invoke *FormGen* and the exact screen responses.

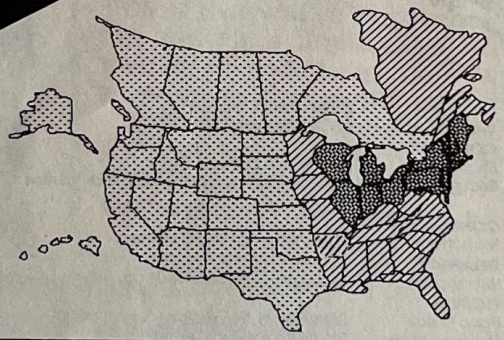
Charlotte Stone, office manager for the Detroit office of the Shaw/Walker Co., has been using a Sanyo computer in her daily work routine since October 1983. Brian Stone has been using a variety of Sanyo computers since May 1983. Both have been involved with computers since their first purchase in July 1978.

NEW JERSEY	
Elizabeth	Dexport Computer Systems
Lake Hopatcong	Jefferson Computer Center
Ocean City	The Computer & Software Store
NEW MEXICO	
Alamogordo	New Horizons Computer Systems
Albuquerque	Page One Newsstand
	Zia Computers
Las Cruces	Zia Computers
NEW YORK	
Hicksville	Innerlogic Computer Center
Rochester	Computer Directions
Wappingers Falls	Software City
NORTH CAROLINA	
Jacksonville	2 M Systems
Mt. Airy	Mountain Computers
NORTH DAKOTA	
Buxton	Systematics
Riverside	Computer Associates
OHIO	
Ashland	Infopro
Canal Fulton	Gemstone Computing
Columbus	The Book Rack
Euclid	Computer Warehouse
Kettering	Electronic Connexion
Mayfield Heights	Programs Unlimited
Marion	Marion Computer Center
North Olmstead	Softwaire Centre International
OKLAHOMA	
Oklahoma City	Merit Computers
Tulsa	Data Station — Tulsa
OREGON	
Beaverton	Creative Computers & Electronics
Eugene	University of Oregon Bookstore
Hillsboro	Quest Enterprises, Inc.
Portland	Megrathea
PENNSYLVANIA	
Butler	Software Corner
Downington	Downington Computer Center
Exton	Software City
Philadelphia	Some Hole In The Wall
SOUTH CAROLINA	
Charleston Heights	Software Haus
Sumter	Micro-Computer Depot
Travelers Rest	Concurrent Technologies Corp.
TENNESSEE	
Knoxville	First Byte Computer Co.
TEXAS	
Bay City	Impulse Computers
San Antonio	Wagner Bros. Computer Store
UTAH	
Provo	Lloyd's Business Machines
VERMONT	
Rutland	Computer Marketing
VIRGINIA	
Alexandria	Alonso Book & Periodicals
Falls Church	Soft Cost
Gloucester Point	Gloucester Electronics
Norfolk	Micro-Enhancements
Springfield	Fairfax Computer Corp.
Virginia Beach	Beach Business Machines
WASHINGTON	
Bellevue	Commercial Computer Systems
Centralia	Centralia Computer Center
College Place	Discount Computer Supply
Olympia	Computer Center
Renton	Puget Sound Electronics
Seattle	Bits & Chips
Tacoma	B & I Magazine & Books
WEST VIRGINIA	
Charleston	Salmon's Diversified Office Services
Dunbar	Wilsumco Computer

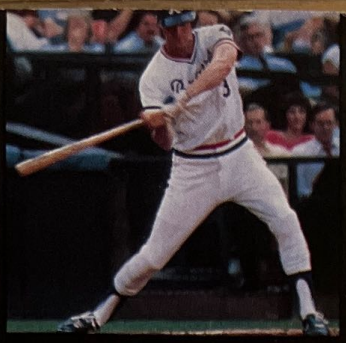
Canadian and Foreign Distributors

ARGENTINA:	
Buenes Aires	ICP, S.A.
AUSTRALIA:	
SYDNEY	
Kingsford	Paris Radio Electronics
VICTORIA	
Melbourne	Computer Biz
CANADA:	
ALBERTA	
Calgary	Tesserac Computer Systems
	The Computer Shop
Edmonton	CompuSoft
	Computique
	Compu-ware
	Disk and Chip
Grand Prairie	LEW Computers
BRITISH COLUMBIA	
Abbottsford	Clearbrook Computer Centre
Richmond	Computer King
Penticton	Corporate Computers
Port Coguitlam	Compulit
Prince George	Computer King
Surrey	Computer King
	The Computer Dept. of London Drugs
Victoria	Odyssey Computer Services
MANITOBA	
Winnipeg	Micro-Mentor, Inc.
NOVA SCOTIA	
Halifax	Atlantic News
ONTARIO	
Burlington	Burlington Business Systems
Guelph	Neutron Computers, Inc.
Kitchener	Home Computer Centre
London	ABA Computer Center
	Town & Country Business Machines
Ottawa	E.T. Wilson Office Equipment LTD.
Peterborough	Computer Support
Scarborough	Scitron
Toronto	Computer Junction
	CP&A
Waterloo	Computer Junction
Windsor	Facts & Figures Computer Service
QUEBEC	
Montreal	Lemond PC
SASKATCHEWAN	
Saskatoon	Memory Lane Computers
ENGLAND	
East Sussex	Molimerx LTD
PANAMA	
Colon	Peikard, Inc.
NEW ZEALAND	
Palmerston North	Viscount Electronics LTD
VENEZUELA	
Maracaibo	Byte Computer Center

ADVERTISERS INDEX



We encourage you to patronize our advertisers — all of whom support the Sanyo line of personal computers. We will appreciate your mentioning **SOFT SECTOR** when you contact these firms.



VCR

THE HOME VIDEO MONTHLY

A funny thing has happened in the home video market. As it gets easier and easier to watch what you want, it gets harder and harder to decide what that might be.

Dozens of new tapes are released into the stores every week, along with hundreds of hours of programming on the network and cable channels. It has reached a point where even the most devoted videophile can use a little help sorting through it all.

Now that help is available.

Inside **VCR** you will find clear, easy-to-read reviews of all the best new releases. You'll find out about little-known videos — what they're about and where to get them. And in the feature articles, you'll find some very entertaining reading about the entertainment business.

Now that you've discovered **VCR**, you won't want to risk missing a single issue. You can save yourself the trouble of walking to the newsstand each month, and save yourself some money at the same time.

Just fill out the attached card and drop it in the mail. Or even easier, call (502) 228-4492 and ask for Sandy. She'll see to it that **VCR** is delivered right to your door each and every month of the year.

Yes, enter my subscription for the next 12 issues of VCR.

At only $15, that's 36 percent off the regular newsstand price.

Name _____

Address _____

City _____ State _____ ZIP _____

☐ My check in the amount of _____ is enclosed.
(In order to hold down costs, we do not bill.)

Charge to: ☐ VISA ☐ MasterCard ☐ American Express

Acct. Number _____ Exp. Date _____

Signature _____

*Subscriptions to **VCR** are $15 a year in the United States. Canadian rate is U.S. $22. Air mail rate elsewhere is U.S. $60. All subscriptions begin with the current issue. Please allow 6 to 8 weeks for first copy. Kentucky residents add 5% sales tax. **U.S. currency *only*, please**. In order to hold down non-editorial costs, we do not bill.

To order by phone (*credit card orders only*) call 1-800-847-0309, 8 a.m. to 5 p.m. EST. For other inquiries call 1-502-228-4492.

Mail to: VCR, The Falsoft Building, P.O. Box 385, Prospect, KY 40059

ACCOUNTING FOR MICROS

$395 Set of Four
$325 Set of Three
$465 Set of Five

ACCOUNTING FOR MICROS is a set of integrated accounting programs which meet professional standards. They're fast and easy to use, with complete instructions. Our manual (shown above) also includes helpful information on bookkeeping and computers.

GENERAL LEDGER $125
Allows up to 1,000 accounts & 1,000 transactions/month. Retains mo/end balances for Last year, This Year and Forecast. Includes Cash Disbursements, Cash Receipts and General Journals. Reports include Balance Sheet, Income Statement, Annual Summaries and Journal Reports.

ACCOUNTS RECEIVABLE $125
Allows up to 2,500 customers and 1,000 invoices per month. Invoicing can access Inventory Module. Keeps customer names and addresses. Invoice prints on plain paper or any pre-printed form. Statements can be printed at any time.

INVENTORY $125
Allows up to 4,000 parts. Keeps 3 month history of unit sales as well as year to date. With AR, can be used as point of sale system (prints invoices, handles cash). Reports include Inventory Value and Stock Report, Internal and Customer Price List.

ACCOUNTS PAYABLE $125
Allows up to 500 vendors and 600 invoices/mo. Records invoices and handwritten checks. Prints computer checks on any pre-printed form. Keeps vendor names and addresses.

PAYROLL $125
Will handle up to 100 employees with eight deductions per employee. Deductions may be determined as fixed dollar amounts or percentages, or referred to a table for automatic look-up. Tax tables are easily entered, or purchased separately. Prints checks and W2's.

SET OF FIVE $465
SET OF FOUR $395
SET OF THREE $325

RUN ON MOST CPM AND MSDOS

Apple CPM	*IBM PC,XT,PC jr,AT*	*Sanyo (all)*
Columbia	*Kaypro (all)*	*Tandy (all)*
Compaq	*Morrow (all)*	*TeleVideo*
Corona	*Osborne (all)*	*Zenith 100 & 150*
Eagle (all)	*Panasonic*	*8" CPM*
Epson QX-10	*Radio Shack CPM*	*Other compatibles*

DEMO DISK $18.00

Try all 5 programs above (GL, AR, AP, IN, PR). Order our DEMO DISK for $18.00 (includes shipping). Condensed versions of the programs give you the "feel" of data entry and access. Includes sample reports and instructions. Specify machine.

TMAN $125
The "Catch-All" program. Files any type of information for quick access. Name or subject oriented with 15 lines of notes per name. Use TMAN as a mailing list, filing system, notebook, etc. Can be used alone or with data from our other programs.
Try ***TMAN DEMO $16***

HOW TO ORDER: Please specify machine and disk format. You can pay by check, by VISA or MasterCard (we need your expiration date and card number), or by UPS COD (add $2.50 COD charge). Our price includes shipping. Minnesota residents, add 6% sales tax). We ship most orders the same day.
or ***ORDER BY PHONE: 612-339-2521***

James River Group
INCORPORATED

(612)339-2521

125 North First Street
Minneapolis, MN 55401